UCRL-TR-220019

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

# Wolter X-Ray Microscope Computed Tomography Ray-Trace Model with Preliminary Simulation Results

J. A. Jackson

March 22, 2006

## Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

# Wolter X-ray Microscope
# Computed Tomography Ray-Trace Model
# with Preliminary Simulation Results

Jessie A. Jackson

September 20, 2004

## TABLE OF CONTENTS

# INTRODUCTION

It is proposed to build a Wolter X-ray Microscope Computed Tomography System in order to characterize objects to sub-micrometer resolution.

Wolter Optics Systems use hyperbolic, elliptical, and/or parabolic mirrors to reflect x-rays in order to focus or magnify an image. Wolter Optics have been used as telescopes and as microscopes. As microscopes they have been used for a number of purposes such as measuring emission x-rays and x-ray fluoresce of thin biological samples.

Standard Computed Tomography (CT) Systems use 2D radiographic images, from a series of rotational angles, acquired by passing x-rays through an object to reconstruct a 3D image of the object.

The x-ray paths in a Wolter X-ray Microscope will be considerably different than those of a standard CT system. There is little information about the 2D radiographic images that can be expected from such a system. There are questions about the quality, resolution and focusing range of an image created with such a system. It is not known whether characterization information can be obtained from these images and whether these 2D images can be reconstructed to 3D images of the object

A code has been developed to model the 2D radiographic image created by an object in a Wolter X-ray Microscope. This code simply follows the x-ray through the object and optics. There is no modeling at this point of other effects, such as scattering, reflection losses etc. Any object, of appropriate size, can be used in the model code. A series of simulations using a number of different objects was run to study the effects of the optics. The next step will be to use this model to reconstruct an object from the simulated data. Funding for the project ended before this goal could be accomplished.

The following documentation includes;

- background information on current X-ray imaging systems,
- background on Wolter Optics,
- description of the Wolter System being used,
- purpose, limitations and development of the modeling code,
- description of the modeling code,
- the results of a number of preliminary imaging simulations,
- recommendations for future Wolter designs and for further modeling studies.

## Current X-ray Imaging Systems

There is often a need to examine, or characterize, the interior of an object without damaging it; this is referred to as non-destructive testing. One method to examine an object non-destructively is to use x-rays.

X-rays are high-energy photons that can pass through objects. However some of the x-ray photons will be absorbed by the materials in the object; different materials will absorb different amounts of x-rays. The quantitative measure of the absorption factor of a material is $\mu$, the *linear attenuation coefficient*. The total amount of x-ray attenuation through a given material is related to the linear attenuation coefficient and the distance the x-rays travel through the material. If $I_o$ is the number of x-ray photons entering an object and $I$ is the number of x-rays exiting, then the attenuation of the x-rays is represented by Equation 1.

$$I = I_o e^{-\int \mu(\ell)d\ell} \qquad [1]$$

The term $\int \mu(\ell)d\ell$ is a line integral where $\mu(\ell)$ represents the value of $\mu$ at each location along the line the x-ray photon travels through an object. Even though the path an x-ray takes through an object can be complicated by scattering of various kinds, the model being considered here will take an idealized approach, in which it will be assumed that the x-rays travel in a straight line through an object.

In a standard x-ray system x-rays pass directly from a source through an object to a detector. It is assumed that the source produces x-rays photons that travel in straight lines called beams. X-ray source configurations typically produce beams that are defined as parallel, fan or cone in shape as shown in Figure 1.

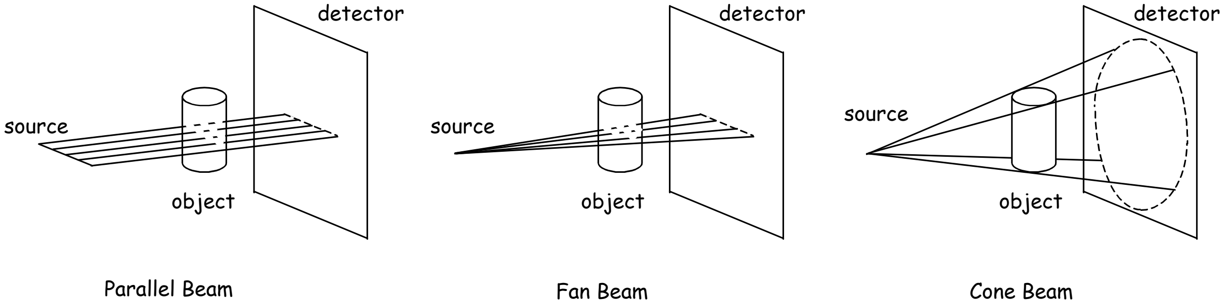


Figure 1- Standard X-ray System Beam Types

The detector records the x-rays that reach it; those which have not been absorbed by the object. Each point on the detector reflects the path along one beam from the source to detector through the object.

The 2D image formed on the detector is called a *projection*. Historically film is used as a detector for x-rays. It is essential for analysis purposes that the image at the detector is digitized. Digitizing entails dividing the image into separate equally sized elements called *pixels* and giving each pixel a value that represents its intensity on the image and storing those values on a computer. If the detector is film the image on the film can be digitized with standard digital scanning software. It is more efficient and accurate to use a detector that collects the X-rays digitally directly. A common digital detector system consists of a scintillator and a CCD camera. A scintillator is a device that converts the x-ray photons to photons of visible light, lower energy photons. Instead of film, a CCD camera uses a silicon wafer that has been segmented into separate photocells. A photon of light hitting a photocell will, by the photoelectric effect, cause an electron to accumulate in the photocell. When the camera shutter closes the counts of electrons in each photocell will be downloaded to create an image that is segmented into pixels.

An x-ray of an object provides a limited amount of information. The image on the detector represents essentially a shadow or a profile of the object as viewed from one direction; a 2D view of a 3D object. However if the object is rotated and a series of digital projections are collected a 3D digital image of the object can be created. The process of creating, or *reconstructing*, a 3D image from a series of 2D digital images is called Computed Tomography (CT).

The objects that are created through CT are digitized. Since objects are 3D, to be digitized, they need to be divided into equally sized 3D elements called *voxels*. The value of a voxel can represent a physical property of the volume of the object corresponding to the voxel location. In this case the object voxels will represent the linear attenuation coefficient. This means that if an x-ray beam traces a path through the object the sum of the length through each voxel along the path multiplied by the value of that particular voxel will represent the attenuation of the x-ray along that path. Let *i* represent the voxels along the beam path, then the attenuation along one beam for the digitized model can be represented by Equation 2.

$$I = I_o e^{-\sum_i \mu_i \ell_i} \qquad [2]$$

This is shown in Figure 2.

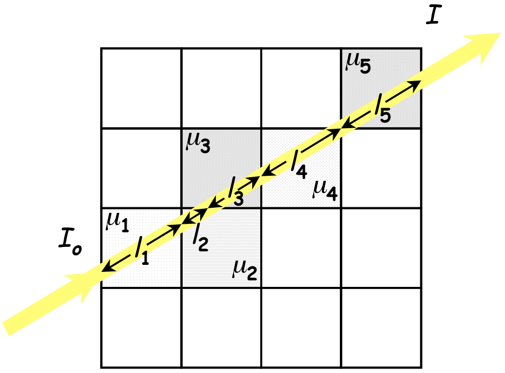


Figure 2 – X-ray Attenuation Along a Path

There are many Computed Tomographic (CT) techniques to create the 3D images. In order to decrease processing speed some of these methods make certain simplifying assumptions about the beam paths and they work well for parallel and fan beam systems. These methods consist of processing the projection data; by filtering, scaling and/or FFTs etc, and then essentially summing the data. These methods include Filtered Backprojection (FBP) and Convoluted Backprojection (CBP). Cone beam systems are much more complicated to deal with. One method for reconstructing objects in a cone-beam system is Constrained Conjugate Gradient Linear Cone (CCG-LCONE). This code has been used only in experimental situations. It is slow and uses a great deal of memory. However it is very flexible and accurate. It uses a forward model of the system, in this case the path of the x-rays through the object (LCONE). It is an iterative method; an initial guess of the object is created and with it the forward model is used to calculate an estimate of the projections. The estimated projection is compared to the real projection values and the difference is used to create a better guess of the value of the object. CCG provides a very efficient method for guessing the next estimate of the object [1].

## Wolter Optics

The optical lenses used in telescopes and microscopes use the refractive properties of visible light photons to redirect the light to a smaller or larger image. X-ray photons do not refract since they pass through objects, albeit with attenuation, so lenses cannot be used to magnify or focus an x-ray image. Mirrors, using the reflective properties of light, can perform the same operations as lenses and an x-ray can be reflected off mirrors made of certain materials. However, since the angle of reflection for x-rays is very small a unique approach is required to avoid huge focal lengths.

In 1952 German physicist Hans Wolter designed three microscope systems for x-rays utilizing a combination of hyperbolic, parabolic and/or elliptical mirrors. The theory behind the designs can be used for either telescopes or microscopes. However until recently the technology has not existed to manufacture the mirrors necessary for the designs. Wolter X-ray Telescopes that work very well have now been designed and built. Wolter X-ray Microscopes designed for various purposes have also been built, however, the requirements of this project are different from existing systems and pose unique challenges.

Figure 3 shows the Wolter design that is being used in this case.

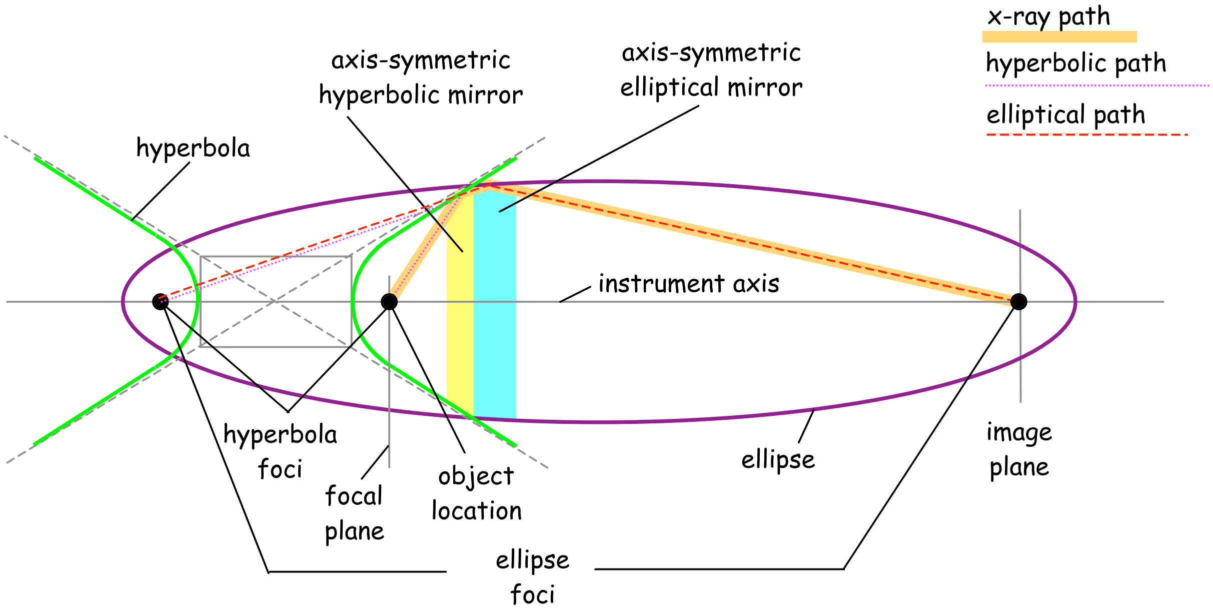


Figure 3 – Wolter Optic Design

This design uses a hyperbolic mirror and an elliptical mirror. The shape of each mirror follows either the hyperbolic or elliptical shape over a small range along the instrument axis. The mirrors are axis-symmetric about the instrument axis as seen in Figure 4, which shows a view of the mirrors from the position of the object looking down the instrument axis toward the mirrors.

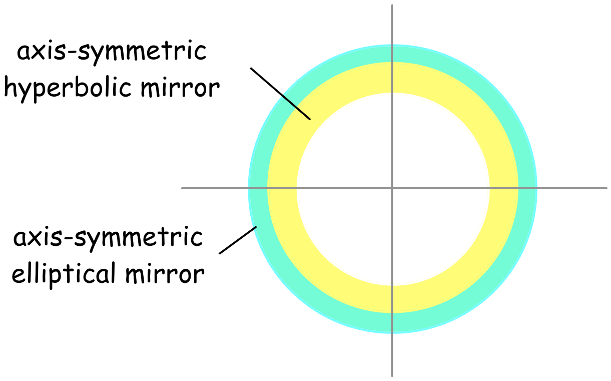


Figure 4 – Wolter Optic Mirrors – Axial View

A key point of the design is that the left focal point of the ellipse and the left focal point of the hyperbola are the same; an object placed at the right hyperbola focal point on the instrument axis, will be in focus at the left hyperbola focal point and therefore in focus at the right ellipse focal point. Also the mirror locations are at areas of low curvature, i.e. small angle of reflection, on both the ellipse and hyperbola.

The image of the object will be reversed around the right elliptical focal point. The image will also be magnified. Points on the focal plane near the instrument axis will appear reversed, magnified, and in focus on the image plane. Points on the focal plane but farther away from the instrument axis will experience off-axis blurring. A point not on the focal plane can be imaged on the image plane but it will not be in focus.

### Wolter X-ray Microscope Design

The Wolter X-ray Microscope System is shown in Figure 5. The source produces the x-rays. Some of the x-rays will pass through the object, then they will be reflected off the mirrors and onto the scintillator. Since the scintillator converts the x-rays to visible light, the scintillator output can be passed through an optical lens to magnify the image before it reaches the CCD camera where the image is acquired. An x-ray stop is placed after the object plane to prevent x-rays from reaching the scintillator without reflecting off the mirrors.

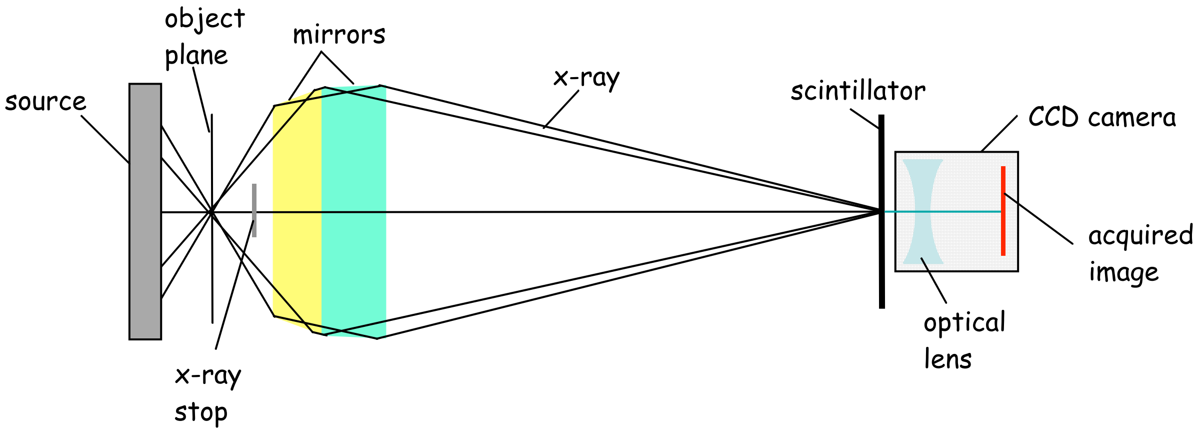


Figure 5 – Wolter X-ray Microscope Design

Five parameters are necessary to specify the system. These parameters affect the function of the Wolter in interrelated ways so the choice of parameter values involves tradeoffs. Some of the issues involved in the tradeoffs include x-ray throughput, off-axis blurring and field of view. A complete discussion of these issues can be found in "Wolter Instrument-Optical Design" [2].

The five parameters needed to specify the system are:

***Magnification***

The system specification is that the resolution of the object must be 0.5um. The resolution of the CCD camera is 18.0 um, so the total magnification must be 36X. For efficiency considerations the scintillator being used in this case has a resolution of 6 um. A lens of 3X between the scintillator and CCD camera will take a 6.0 um signal at the scintillator and magnify it to be seen as 18.0 um at the CCD camera.

Therefore the image of the object, with a resolution of 0.5um, must be magnified by 12X by the Wolter Optic before reaching the scintillator.

***Instrument Length***

The instrument length, ***L***, is the distance from the object plane to the image plane.

***Hyperbola Mirror Acceptance Angle***

In order to form a statistically valid image a certain number of x-ray photons must reach the scintillator. The number of photons will depend on the number produced by the source and the percentage of those that are passed through the mirrors. There are efficiency losses associated with the mirrors due to scattering and other effects. However the shape of the mirrors decide the maximum possible throughput. The angle $\boldsymbol{\Delta\theta_c}$ in Figure 6 is a way of specifying the throughput; the larger this angle the more x-rays will be passed through.

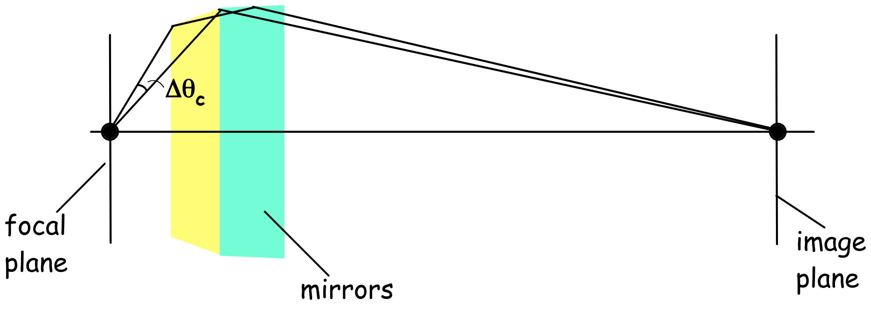


Figure 6 – Hyperbolic Mirror Acceptance Angle

### *Maximum Collection Angle*

The angle $\boldsymbol{\theta_{cf}}$ in Figure 7 is defined as the maximum collection angle. This angle has an influence on the off-axis blur at the image plane and the x-ray throughput.

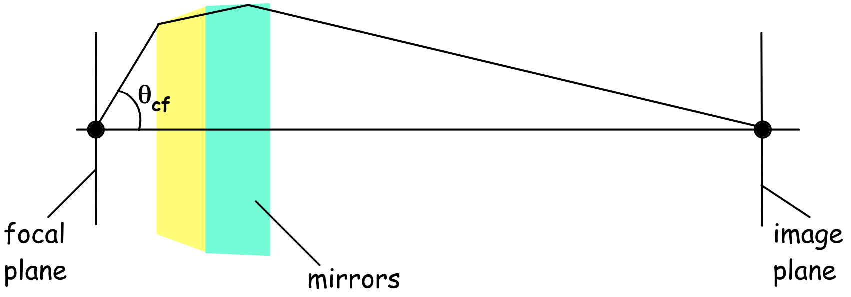


Figure 7 – Maximum Collection Angle

### *Hyperbola Minor Axis*

A hyperbola is specified by the parameters shown in Figure 12. The value of the minor axis, $\mathbf{b_c}$, is needed to specify the Wolter X-ray microscope.

The values selected for these parameters are:

M = 12

L = 5000 mm

$\theta_{cf}$ = 5.6 degrees

$\Delta\theta_c$ = 0.179 degrees

$b_c$ = 14.58 mm

# CODE DEVELOPMENT

## Purpose and Limitations

The main purpose of the code is to create a ray-trace model of the x-rays through an object and the Wolter Optics in order to determine what an image from such a system can be expected to look like. The model will not include any effects other than the ray-tracing; there is no modeling of scattering, diffraction, reflection losses etc. So the images that are created will be rather idealistic. However a Wolter System has not been used in this fashion before so simulation results from the model should provide interesting information on what kind of images to expect, possible design improvements and ways to use the images to characterize the object.

An additional goal is to investigate possible ways of reconstructing the object using the simulated images.

## Code Development - Accomplished

This code originated as a means to study the x-ray paths during the design process, without an object present in the system. It was originally written in IDL. IDL is an excellent system to use when processing data arrays, which the original application was doing. When the time came to add object attenuation it was decided to modify this existing code in order to get some basic results quickly. This version of the code produces the desired initial results; in fact the simulations presented later in this paper are all created with it.

The original code traces the x-rays from the source to the detector. Since so many x-rays do not even reach the mirrors this proves to be an inefficient way to produce images. Also artifacts in the image occur due to the digitization process when ray-tracing from the source to detector. So the first step in modifying the code was to reverse the direction of the ray-trace so it is detector to source. Then the calculation of the attenuation of an x-ray through an object was added.

Since a single image of an object provides very limited information about the object, traditional CT uses multiple rotational views to characterize objects more completely. The Wolter X-ray Microscope initial hardware design includes plans for a stage for translating the object along the instrument axis. Therefore the code was modified to allow simulation of images from multiple positions of the object along the instrument axis.

A series of simulations was done; they are described in a later section.

At this point it was realized that the processing in the code is no longer done with arrays and the code is at the point where the implementation in IDL is much slower than a version in C. Also the existing reconstruction codes in NDE for standard CT (RECON) are in C and it will be easier to incorporate the Wolter ray-trace code into them if it is in C. The RECON system also has a standard I/O package for both data and processing parameters. Therefore it was decided that the current version of the Wolter ray-trace code should be converted to C and the parameter and data I/O for RECON should be included. This process was not finished when the funding for the project was stopped.

### Code Development – Future Work

After the code is converted to C and the RECON interface is added the next step will be to incorporate the Wolter ray-trace model into the RECON reconstruction code CCG. CCG is an iterative reconstruction method that uses a ray-trace model for the system. Then it will be possible to see if the images from a Wolter system will provide enough information to perform a reconstruction of the object.

The final steps will be to add object rotation to the model and simulate images and reconstructions.

To summarize, the past and future development for the code is as follows:

- ✓ reverse ray-trace; detector to source,
- ✓ add object attenuation,
- ✓ add translation of object along instrument-axis,
- ✓ produce simulations,
- ➢ convert code to C, with RECON type data I/O,
- o incorporated ray-trace model into CCG,
- o perform reconstructions on simulated images,
- o add object rotation,
- o perform simulations with object rotations,
- o perform reconstruction with simulations from rotated objects.

# CODE DESCRIPTION

## Coordinate System

The model code uses a 3D coordinate system; the object location, which is the intersection of the focal plane and the instrument axis, is the origin of the coordinate system, the **z**-axis extends from the origin along the instrument axis to the scintillator. The **y**-axis extends vertically from the origin and **x**-axis extends into the paper.

The ray path will be defined by the current location **p** = ($p_x$, $p_y$, $p_z$) and the unit direction of the ray **d** = ($d_x$, $d_y$, $d_z$). The coordinate system is shown in Figure 8.

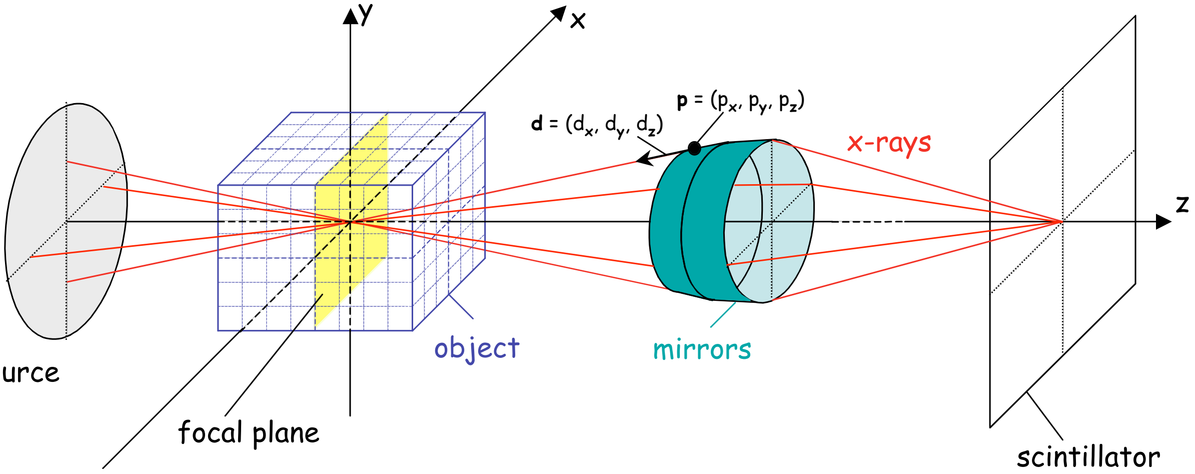

y
x
p = (p<sub>x</sub>, p<sub>y</sub>, p<sub>z</sub>)
d = (d<sub>x</sub>, d<sub>y</sub>, d<sub>z</sub>)
x-rays
z
source
object
mirrors
focal plane
scintillator

Figure 8 – Wolter Ray-Trace Model Code Coordinate System

## Nomenclature

The parameters used in describing the code have the following nomenclature. Angles use the Greek alphabet, **θ** and **γ**, locations along the z-axis use **z**. The mirrors are axis-symmetric about the **z**-axis. At times parameters will be determined using a radial distance **r**, which is perpendicular to the z-axis, where $\mathbf{r}^2 = \mathbf{x}^2 + \mathbf{y}^2$. **Δ** represents a difference. The hyperbolic mirror is referred to as **c**, the elliptical mirror will referred to as **d**. The front edge of a mirror is indicated by **f**, and the back edge by **b,** and **i** will indicate the intersection between the mirrors. (The back edge of mirror **c**, the front edge of mirror **d** and **i** all indicate the same location.) So a parameter representing the z-axis location of the front edge of the hyperbolic mirror will be $\mathbf{z_{cf}}$ and the distance at that point from the z-axis to the mirror is given by $\mathbf{r_{cf}}$.

## Parameter and Data I/O

Non-Destruction Evaluation (NDE) at LLNL has file formats for handling data and parameter information files for their CT systems. Even though the Wolter model code is not part of the suite of support software it uses the same file formats. Data files are saved as binary files with the tag ".sdt". An associated file with the type and dimensions of the data file is saved as an ASCII file with the same name as the data file and the tag ".spr". Data files containing the linear attenuation coefficients of an object being studied will begin with the letter “o”, ie. o0001.sdt and o0001.spr. Data files containing the resultant projections will begin with the letter “p”, i.e. p0001.sdt and p0001.spr.

When a CT code is run the user can define certain run time parameters. These user defined parameters are defined in a file with a tag of “.sct”. The format of the SCT files is a ‘-‘ followed by a parameter name, then whitespace, then parameter value. The CT codes read in the designated SCT file and use the parameters to determine how to process the run. The parameter names used in the SCT files are listed here; an explanation of the various parameters will be given in the following sections.

**SCT File Parameters**

| | |
|---|---|
| **-ofile** | object file name (input) |
| **-pfile** | projection file name (output) |
| **-ops.power_flag** | a parameter for the reflection power off the mirrors (NOT USED) |
| Source information | |
| **-src.ctr.x** | x location of the center of the source (mm) |
| **-src.ctr.y** | y location of the center of the source (mm) |
| **-src.ctr.z** | z location of the center of the source (mm) |
| **-src.ctr.r** | radius of source (mm) |
| Scintillator information | |
| **-scint.nang.3** | number of ray paths across mirror D |
| **-scint.nang.4** | number of rotational ray paths on mirror D |
| **-scint.sz.x** | size of scintillator pixel in x direction (mm) |
| **-scint.sz.y** | size of scintillator pixel in y direction (mm) |
| **-scint.sz.z** | size of scintillator pixel in z direction (mm) |
| **-scint.ctr.x** | x location for center of scintillator (mm) |
| **-scint.ctr.y** | y location for center of scintillator (mm) |
| **-scint.ctr.z** | z location for center of scintillator (mm) |
| **-scint.ngrid.x** | number scintillator of pixels in x direction |
| **-scint.ngrid.y** | number scintillator of pixels in y direction |
| Object information | |
| **-obj.sz.x** | size of object voxel in x direction (mm) |
| **-obj.sz.y** | size of object voxel in y direction (mm) |
| **-obj.sz.z** | size of object voxel in z direction (mm) |
| **-obj.ctr.x** | x location of center of object (relative to ops.sctr.x) (mm) |
| **-obj.ctr.y** | y location of center of object (relative to ops.sctr.y) (mm) |
| **-obj.ctr.z** | z location of center of object (relative to ops.sctr.z) (mm) |
| **-obj.ngrid.x** | number of object voxels in x direction |
| **-obj.ngrid.y** | number of object voxels in y direction |
| **-obj.ngrid.z** | number of object voxels in z direction |
| Run information | |
| **-ops.sctr** | location of object center on z-axis for first projection (mm) |
| **-ops.dctr** | step size for object along z-axis for multiple projections (mm) |
| **-ops.nctr** | number of projections |

At the end of a run the CT codes add information about the run to the SCT file to provide run tracking capabilities. The Wolter model code does not do this at this time. It produces a ".txt" file with run information. This comes from the previous versions of the code and will be changed in future version of this code.

**Mirror Parameters**

In order to calculate the ray path intersection with and reflection off of the mirrors certain mirror geometry parameters must be determined. These parameters can be calculated from the five design variables described previously. They are independent of the ray path so they need be calculated only once.

The parameters can be divided into five groups:

- the location of the intersection of the two mirrors,
- the hyperbola geometry,
- the ellipse geometry,
- the hyperbolic mirror dimensions,
- the elliptical mirror dimensions.

Figure 9 shows an overview of the interrelation between the geometries of the two mirrors and Figure 10 shows the mirror dimension parameters. The known design variables are in red.

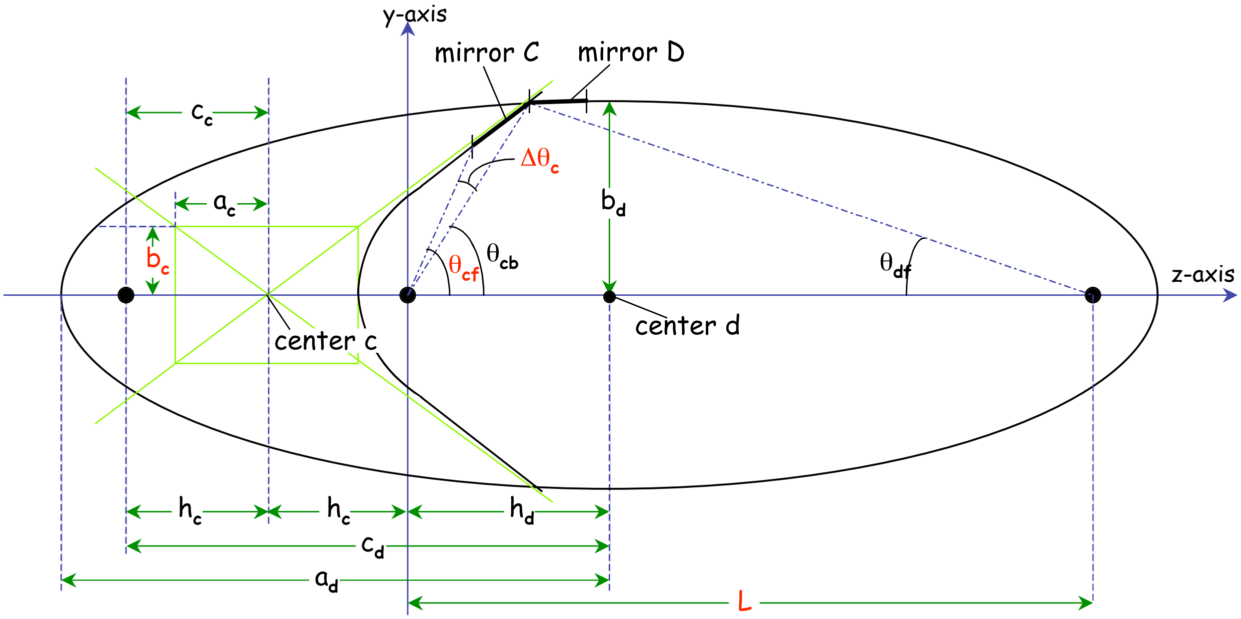


Figure 9 – Mirror Parameter Relationships

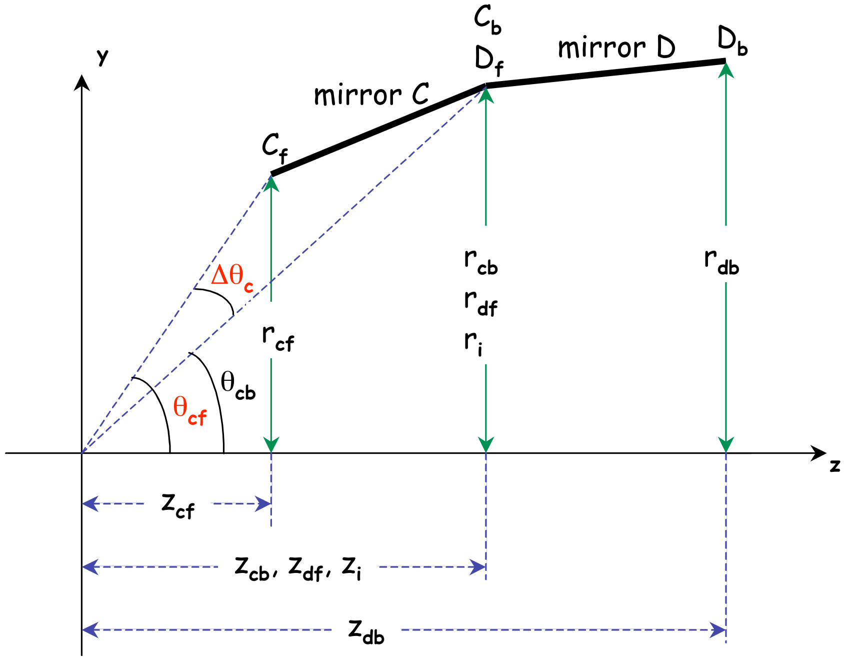


Figure 10 – Mirror Dimension Parameters

### *Mirror Intersection Location*

First of all, by examining Figure 10 it is obvious that $\boldsymbol{\theta}_{\mathbf{cb}} = \theta_{\mathbf{cf}} - \Delta\theta_{\mathbf{c}}$

The next parameters that need to be calculated are $\mathbf{r_i}$ and $\mathbf{z_i}$, these describe the location of the intersections of the two mirrors. Figure 11 shows the variables needed in determining $\mathbf{r_i}$ and $\mathbf{z_i}$.

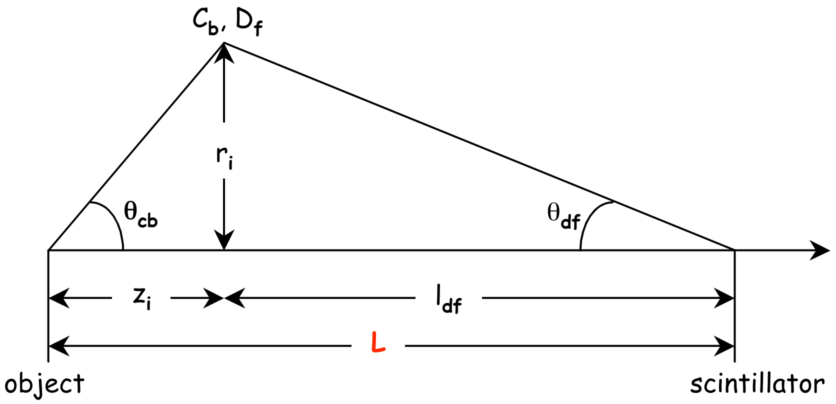


Figure 11 – Variables for Determining $\mathbf{r_i}$ and $\mathbf{z_i}$

**L**, and **M** are known and by definition

$$M = \frac{\sin\theta_{cb}}{\sin\theta_{df}} \tag{3a}$$

Therefore $\theta_{\mathbf{df}}$ can be determined as

$$\boldsymbol{\theta_{df}} = \sin^{-1}\left(\frac{\sin\theta_{cb}}{M}\right) \tag{3b}$$

Then

$$\frac{\tan\theta_{df}}{\tan\theta_{cb} + \tan\theta_{df}} = \frac{\dfrac{r_i}{l_{df}}}{\dfrac{r_i}{z_i} + \dfrac{r_i}{l_{df}}} = \frac{z_i}{L} \tag{4a}$$

Therefore

$$z_i = \boldsymbol{z_{cb}} = \boldsymbol{z_{df}} = L * \frac{\tan\theta_{df}}{\tan\theta_{cb} + \tan\theta_{df}} \tag{4b}$$

Finally

$$\boldsymbol{r_i} = \boldsymbol{r_{cb}} = \boldsymbol{r_{df}} = z_i * \tan\theta_{cb} \tag{5}$$

### *Hyperbola Geometry*

The parameters needed to describe a hyperbola are shown in Figure 12.

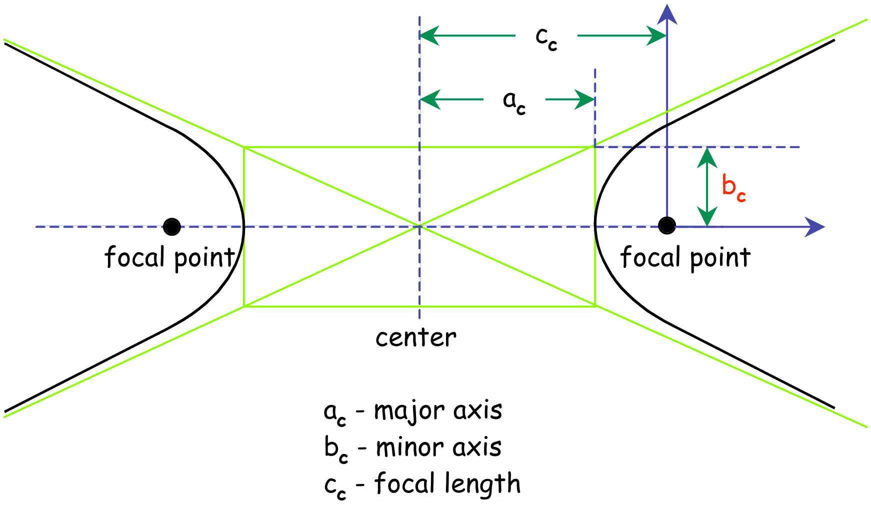


Figure 12 – Hyperbola Geometry Parameters

The equations to describe a hyperbola are

$$b_c^2(z - z_o)^2 - a_c^2 r^2 = a_c^2 b_c^2 \tag{6a}$$

$$a_c^2 + b_c^2 = c_c^2 \tag{6b}$$

$\mathbf{z_o}$ represents the location of the hyperbola center on the z-axis which in this case will be $\mathbf{-c_c}$ as seen in Figure 9. $\mathbf{b_c}$ is a given design variable. The point, or circle, defined by the parameters $\mathbf{r_i}$ and $\mathbf{z_i}$, that were just calculated, is the intersection between the two mirrors so by definition it defines a point, or circle, on the hyperbola. So if $\mathbf{r_i}$, $\mathbf{z_i}$ and $\mathbf{z_o} = \mathbf{-c_c}$ are substituted into Equation 5 the result is

$$b_c^2\left(z_i + c_c\right)^2 - a_c^2 r_i^2 = a_c^2 b_c^2 \qquad [7]$$

The only two knowns are $\mathbf{a_c}$ and $\mathbf{c_c}$ so they can be solved for using Equations 6b and 7 and the result is

$$\boldsymbol{a_c} = \sqrt{\frac{-B + \sqrt{B^2 - 4AC}}{2A}} \qquad [8a]$$

where

$$\begin{aligned} A &= r_i^4 \\ B &= -2r_i^2 b_c^4 - 2r_i^2 b_c^2 z_i^2 - 4z_i^2 b_c^4 \\ C &= b_c^8 - 2b_c^6 z_i^2 + b_c^4 z_i^4 \end{aligned} \qquad [8b]$$

Then

$$\boldsymbol{c_c} = \sqrt{a_c^2 + b_c^2} \qquad [8c]$$

***Ellipse Geometry***

The parameters for an ellipse are shown in Figure 13.

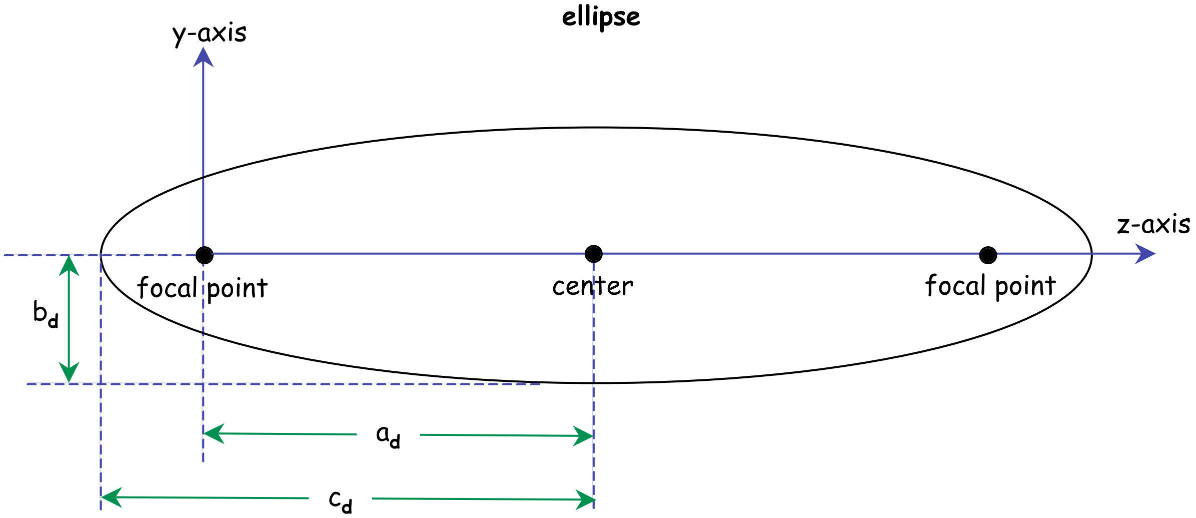


Figure 13 – Ellipse Geometry Parameters

Looking back to Figure 9 it can be seen that

$$\begin{aligned} \boldsymbol{h_c} &= c_c \\ \boldsymbol{c_d} &= \frac{L + 2h_c}{2} \\ \boldsymbol{h_d} &= \frac{L - 2h_c}{2} \end{aligned} \qquad [9]$$

The equations to describe an ellipse are

$$b_d^2\left(z - z_o\right)^2 + a_d^2 r^2 = a_d^2 b_d^2 \qquad [10a]$$

$$a_d^2 = b_d^2 + c_d^2 \qquad [10b]$$

In this case **z<sub></sub>**

In this case $\mathbf{z_o} = \mathbf{h_d}$ and again $\mathbf{z_i}$ and $\mathbf{r_i}$ are used since they are also points on the ellipse giving the equation

$$b_d^2\left(z_i - h_d\right)^2 + a_d^2 r_i^2 = a_d^2 b_d^2 \qquad [11]$$

Using Equations 10b and 11, $\mathbf{b_d}$ and $\mathbf{a_d}$ can be solved for; the results are

$$\boldsymbol{b_d} = \sqrt{\frac{-B + \sqrt{B^2 - 4AC}}{2A}} \qquad [12a]$$

where

$$A = 1.0$$
$$B = c_d^2 - (z_i - h_d)^2 - r_i^2 \qquad [12b]$$
$$C = -r_i^2 c_d^2$$

then

$$\boldsymbol{a_d} = \sqrt{b_d^2 + c_d^2} \qquad [12c]$$

### *Hyperbolic Mirror Dimensions*

Now to solve for $\mathbf{r_{cf}}$ and $\mathbf{z_{cf}}$. For Figure 10 it can be seen that

$$\tan\theta_{cf} = \frac{r_{cf}}{z_{cf}} \qquad [13]$$

Since $\mathbf{r_{cf}}$ and $\mathbf{z_{cf}}$ are on the hyperbola they can be used in the hyperbola formula as follows

$$r_{cf} = z_{cf}\tan\theta_{cf}$$
$$b_c^2\left(z_{cf} - h_c\right)^2 - a_c^2 z_{cf}^2 \tan^2\theta_{cf} = a_c^2 b_c^2 \qquad [14]$$

Then solve for $\mathbf{z_{cf}}$

$$\boldsymbol{z_{cf}} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \qquad [15a]$$

where

$$A = b_c^2 - a_c^2 \tan^2\theta_{cf}$$
$$B = -2b_c^2 h_c^2 \qquad [15b]$$
$$C = b_c^2 h_c^2 - a_c^2 b_c^2$$

and

$$\boldsymbol{r_{cf}} = z_{cf}\tan\theta_{cf} \qquad [15c]$$

### *Elliptical Mirror Dimensions*

Finally $\mathbf{z_{db}}$ can be calculated. $\mathbf{r_{db}}$ could be determined but it is not needed. Examining Figure 14 it can be seen that the line defined by the angle γ goes through both ($\mathbf{r_{cf}}$, $\mathbf{z_{cf}}$) and ($\mathbf{r_{db}}$, $\mathbf{z_{db}}$).

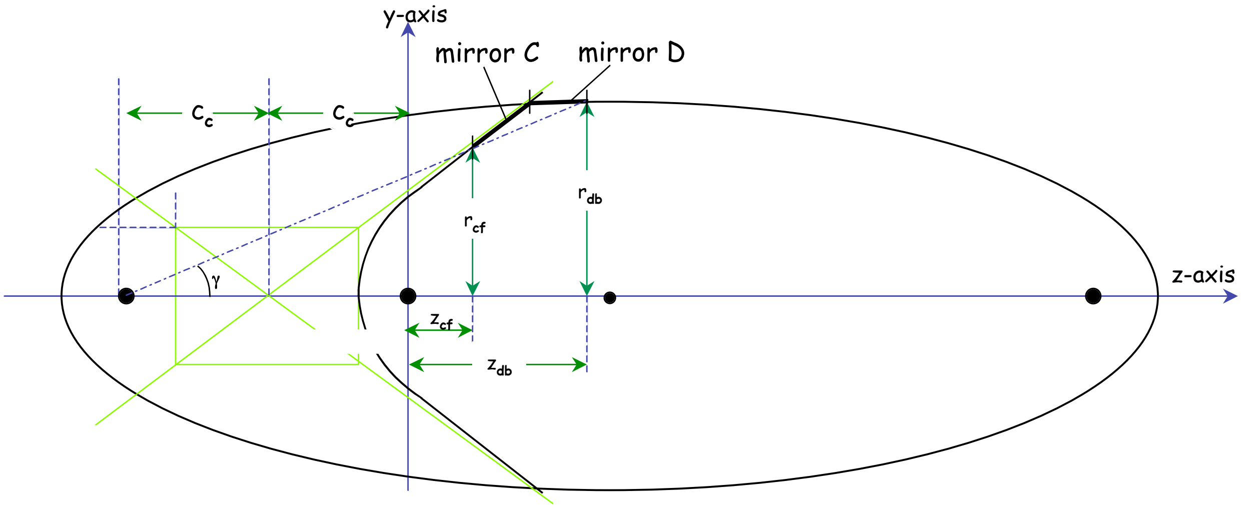


Figure 14 – Elliptical Mirror Dimensions

This means that
$$\tan\gamma = \frac{r_{cf}}{z_{cf} + 2c_c} = \frac{r_{db}}{z_{db} + 2c_c} \quad [16a]$$

Since (**r**$_{\mathbf{cf}}$**, z**$_{\mathbf{cf}}$) are known tanγ can be calculated and then

$$r_{db} = \left(z_{db} + 2c_c\right)\tan\gamma \quad [16b]$$

(**r**$_{\mathbf{db}}$**, z**$_{\mathbf{db}}$) are the end points of the elliptical mirror, so using the ellipse formula and substituting **r**$_{\mathbf{db}}$ from Equation xx in gives the following

$$b_d^2\left(z_{db} - h_d\right)^2 + a_d^2\left(z_{db} + 2c_c\right)^2\tan^2\gamma = a_d^2 b_d^2 \quad [17]$$

Now **z**$_{\mathbf{db}}$ can be solved as
$$z_{db} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad [18a]$$

where
$$\begin{aligned} A &= b_d^2 + a_d^2\tan^2\gamma \\ B &= -2b_d^2 h_d + 4a_d^2 c_c \tan^2\gamma \\ C &= h_d^2 + 4a_d^2 c_c \tan^2\gamma - a_d^2 b_d^2 \end{aligned} \quad [18b]$$

## Ray-tracing

The code essentially traces the ray path backward from the scintillator to the source, as see in Figure 15. The scintillator (1) is considered the image plane in the model. The path then goes to the elliptical mirror (2), the hyperbolic mirror (3) and then onto the source (4). If the ray path doesn’t hit the mirrors or the source there is no need to calculate the attenuation through the object. If it is determined that the ray path reaches the source the intersection of the ray path on the front face (5) and the back face (6) of the object are determined and attenuation on that path through the object will be calculated.

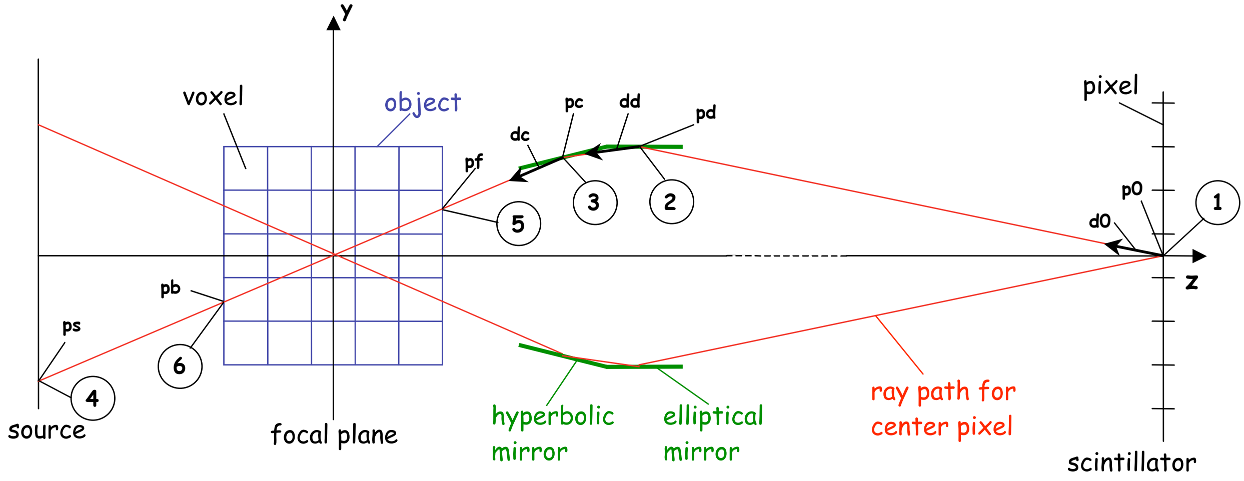


Figure 15 – Ray Path through Wolter Model

### *Reflection - Normal Dot Incidence*

An important part of the ray-tracing is calculating the change of direction of the ray when it reflects off the mirrors. The basic part of this is normal dot incidence. As seen in Figure 16 a given surface will have a vector normal to it, **n**. The incoming vector is $\mathbf{V_1}$, which is known, and the reflected vector is $\mathbf{V_2}$.

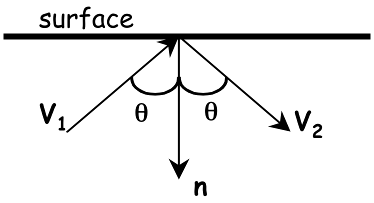


Figure 16 – Reflection, Normal Dot Incidence

Beginning with the definition of a dot product

$$\mathbf{V}_1 \bullet \mathbf{V}_2 = -|\mathbf{V}_1||\mathbf{V}_2|\cos 2\theta \qquad [19]$$

The angle between $\mathbf{V_1}$ and $\mathbf{V_2}$ is 2**θ**. The sign is an indication of the direction of the vectors. The equation for $\mathbf{V_2}$ in terms of $\mathbf{V_1}$ can be found

$$\mathbf{V}_2 = \frac{2(\mathbf{V}_1 \bullet \mathbf{n})}{|\mathbf{V}_1|}\mathbf{n} + \frac{\mathbf{V}_1}{|\mathbf{V}_1|} \qquad [20]$$

***Scintillator* – (1)**

The code starts at the scintillator and loops through all the pixels. The pixel loops on each axis, x and y, will start at the smallest (most negative) x and y locations; the y-axis pixels will loop the fastest, as seen in Figure 17. All the rays for a pixel are processed before going on the next pixel.

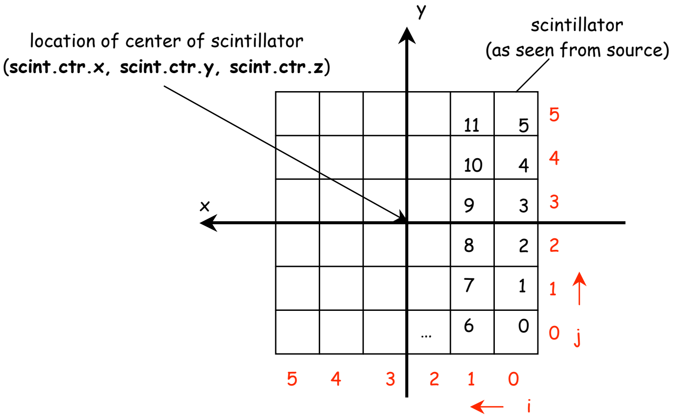

y
scintillator
(as seen from source)
location of center of scintillator
(scint.ctr.x, scint.ctr.y, scint.ctr.z)
x
11
5
10
4
9
3
8
2
7
1
...
6
0
5
4
3
2
1
0
j
5
4
3
2
1
0
i

Figure 17 – Scintillator Pixel Processing Order

The first step in following the path is to calculate the location of the pixel, the start of the path, the vector **p0**. The location of the center of the scintillator (scint.ctr.x, scint.ctr.y, scint.ctr.z), the size of each pixel (scint.sz.x, scint.sz.y) and the number of pixels in each direction (scint.ngrid.x, scint.ngrid.y) are all defined in the SCT file. Using these parameters **p0** can be calculated for each value of *i* and *j*.

$$\begin{aligned}
\mathrm{p0}_{\mathrm{x}} &= scint.ctr.x + (scint.sz.x(i - \frac{scint.ngrid.x}{2} + 0.5) \\
\mathrm{p0}_{y} &= scint.ctr.y + (scint.sz.y(j - \frac{scint.ngrid.y}{2} + 0.5) \qquad [21] \\
\mathrm{p0}_{\mathrm{z}} &= scint.ctr.z
\end{aligned}$$

In a standard CT system there is one ray from the source to each pixel, in a Wolter System there are many rays from mirror D reflecting to each pixel. A schematic profile view of this effect is shown in Figure 18.

In order to represent this in a digital model the paths of a fixed number of rays will be processed for each pixel. Each path will be followed, if the path reaches the source the attenuation through the object will be determined and the attenuated value of that ray will be added to the pixel.

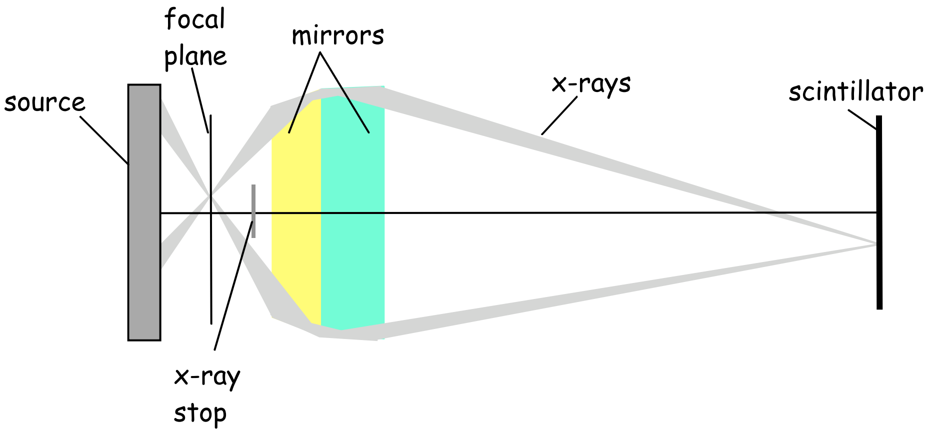


Figure 18 – Multiple Ray Paths per Pixel

## *Mirror D – Ellipse –* (2)

### *Ray Intersection*

There is no need to calculate the intersection of the ray path with this mirror because the mirror is used to define the initial path. For a given pixel a fixed number of rays will be processed. The paths are defined by the pixel location at one end and a location on the mirror at the other end. The rays will be evenly spaced over the mirror. The user specifies the ray locations with two parameters as shown in Figure 19, the number of rotational ray positions, scint.nang.a2, and the number step along the z-axis, scint.nang.a1.

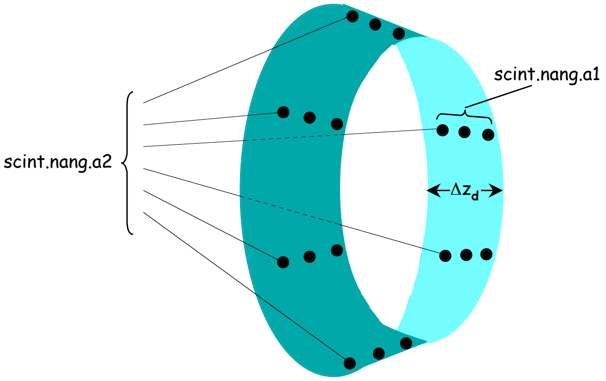


Figure 19 – Ray Path Definition Parameters

Figure 20 shows the relation between the points defined in Figure 19 and the ray paths to be calculated.

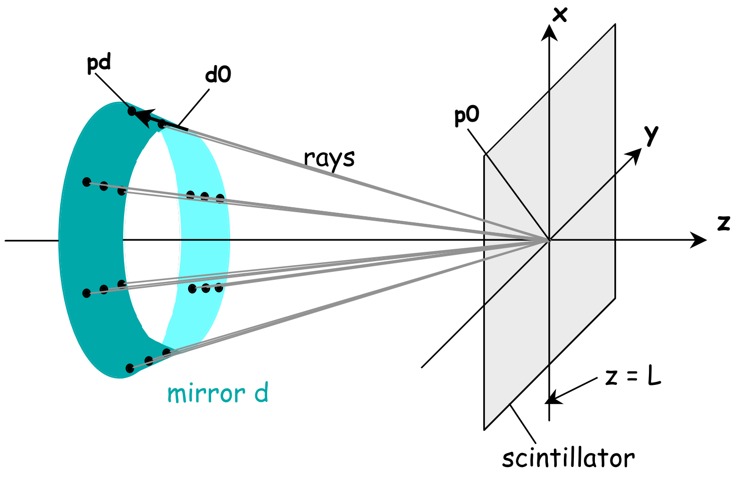


Figure 20 – Ray Path for Scintillator to Elliptical Mirror

The location of pixel, **p0**, is known; the direction vector **d0** must be determined.

First calculate the ray spacing on the mirror

$$\Delta\theta dr = \frac{360^{\circ}}{scint.nang.a2} \qquad \Delta dz = \frac{z_{db} - z_{df}}{scint.nang.a1} \qquad [22]$$

Then using the ellipse equation, Equation 10, the polar coordinate form of the location of points on the mirror can be determined

$$pdz_i = z_{df} + \frac{\Delta dz}{2} * i \qquad pdr_i = b_d^2 - \frac{b_d^2}{a_d^2}\left(pdz_i - h_d\right)^2 \qquad \text{where } i = 0, 1, \ldots \text{ scint.nang.a1} - 1 \qquad [23a]$$

$$\theta dr_j = \Delta\theta dr * j \qquad \text{where } j = 0, 1, \ldots \text{scint.nang.a2} - 1 \qquad [23b]$$

Then the vector **pd** can be calculated

$$\text{pd}_x = pdr_i \cos\theta dr_j \qquad \text{pd}_y = pdr_i \sin\theta dr_j \qquad \text{pd}_z = pdz_i \qquad [24]$$

Finally the direction vector **d0** can be determined

$$\begin{aligned} lcd &= \sqrt{\left(\mathrm{p0_x} - \mathrm{pd_x}\right)^2 + \left(\mathrm{p0_y} - \mathrm{pd_y}\right)^2 + \left(p0_z - pd_z\right)^2} \\ \mathrm{d0_x} &= \frac{\mathrm{p0_x} - \mathrm{pd_x}}{lcd} \\ \mathrm{d0_y} &= \frac{\mathrm{p0_y} - \mathrm{pd_y}}{lcd} \\ \mathrm{d0_z} &= \frac{\mathrm{p0_z} - \mathrm{pd_z}}{lcd} \end{aligned} \qquad [25]$$

### *Normal to Ellipse*

The 3D equation for an ellipse is

$$b_d^2\left(z - h_d\right) + a_d^2 x^2 + a_d^2 y^2 = a_d^2 b_d^2 \qquad [26]$$

The normal will be the gradient of the curve defined by Equation 26 and the location of **pd**. The resulting normalized vector will be **nd** = ($\mathrm{nd_x}$, $\mathrm{nd_y}$, $\mathrm{nd_z}$).

$$\begin{aligned} length &= \sqrt{a_d^4\,\mathrm{pd_x^2} + a_d^4\,\mathrm{pd_y^2} + b_d^4\left(\mathrm{pd_z} - h_d\right)^2} \\ \mathrm{nd_x} &= \frac{-a_d^2\,\mathrm{pd_x}}{length} \\ \mathrm{nd_y} &= \frac{-a_d^2\,\mathrm{pd_y}}{length} \\ \mathrm{nd_z} &= \frac{-b_d^2\left(\mathrm{pd_z} - h_d\right)}{length} \end{aligned} \qquad [27]$$

### *Reflected Direction*

Now the direction vector leaving the ellipse, **dd,** needs to be determined. Referring to Equation 20 the arrival direction vector $\mathbf{V_1}$ will be **d0**. **d0** is already a unit vector so $|\mathbf{V_1}| = 1.0$. The departure vector $\mathbf{V_2}$ will be **dd**. So using the normal vector to an ellipse shown in Equation xx **dd** is

$$\begin{aligned} \mathrm{dd_x} &= \mathrm{d_x} + 2.0 * \left(\mathbf{d0} \bullet \mathbf{nd}\right)\mathrm{ne_x} \\ \mathrm{dd_y} &= \mathrm{d_y} + 2.0 * \left(\mathbf{d0} \bullet \mathbf{nd}\right)\mathrm{ne_y} \\ \mathrm{dd_z} &= \mathrm{d_z} + 2.0 * \left(\mathbf{d0} \bullet \mathbf{nd}\right)\mathrm{ne_z} \\ &\textit{where} \\ \mathbf{d0} \bullet \mathbf{nd} &= \mathrm{d0_x}\,\mathrm{nd_x} + \mathrm{d0_y}\,\mathrm{nd_y} + \mathrm{d0_z}\,\mathrm{nd_z} \end{aligned} \qquad [28]$$

## *Mirror C* – Hyperbola – (3)

### *Ray Intersection*

The intersection of the ray with the hyperbola must be calculated. The current position of the ray is on the ellipse at position vector **pd** and the direction vector is **dd**. If the distance from the ellipse to hyperbola along the ray is the unknown variable *lcd* then the intersection point, **pc**, will be

$$\begin{aligned} &\mathbf{pc} = \mathbf{pd} + lcd * \mathbf{dd} \\ &or \\ &\mathrm{pc}_\mathrm{x} = \mathrm{pd}_\mathrm{x} + lcd * \mathrm{dd}_\mathrm{x} \\ &\mathrm{pc}_\mathrm{y} = \mathrm{pd}_\mathrm{y} + lcd * \mathrm{dd}_\mathrm{y} \\ &\mathrm{pc}_\mathrm{z} = \mathrm{pd}_\mathrm{z} + lcd * \mathrm{dd}_\mathrm{z} \end{aligned} \tag{29}$$

The 3D equation for a hyperbola is

$$b_c^2 (z - h_c)^2 - a_c^2 x^2 - a_c^2 y^2 = a_c^2 b_c^2 \tag{30}$$

If ($\mathrm{pc}_\mathbf{x}$, $\mathrm{pc}_\mathbf{y}$, $\mathrm{pc}_\mathbf{z}$) is substituted for (x, y, z) in Equation 30 the variable *lcd* is the only unknown. The value of *lcd* is then

$$lcd = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{31a}$$

where
$$\begin{aligned} A &= b_c^2\,\mathrm{dd}_\mathrm{z}^2 - a_c^2\,\mathrm{dd}_\mathrm{x}^2 - a_c^2\,\mathrm{dd}_\mathrm{y}^2 \\ B &= 2b_c^2\,\mathrm{dd}_\mathrm{z}\mathrm{pd}_\mathrm{z} - 2h_c b_c^2\,\mathrm{dd}_\mathrm{z} - 2a_c^2\,\mathrm{dd}_\mathrm{x}\mathrm{pd}_\mathrm{x} - 2a_c^2\,\mathrm{dd}_\mathrm{y}\mathrm{pd}_\mathrm{y} \\ C &= -2b_c^2 h_c\,\mathrm{pd}_\mathrm{z} - b_c^2 h_c^2 - a_c^2\,\mathrm{pd}_\mathrm{x}^2 - a_c^2\,\mathrm{pd}_\mathrm{y}^2 - a_c^2 b_c^2 \end{aligned} \tag{31b}$$

### *Normal to Hyperbola*

The normal to the hyperbola, **nc**, will be the gradient of the curve defined by Equation 30. The resulting normalized vector will be

$$\begin{aligned} length &= \sqrt{a_c^4\,\mathrm{pc}_\mathrm{x}^2 + a_c^4\,\mathrm{pc}_\mathrm{y}^2 + b_c^4(\mathrm{pc}_\mathrm{z} - h_c)^2} \\ \mathrm{nc}_\mathrm{x} &= \frac{-a_c^2\,\mathrm{pc}_\mathrm{x}}{length} \\ \mathrm{nc}_\mathrm{y} &= \frac{-a_c^2\,\mathrm{pc}_\mathrm{y}}{length} \\ \mathrm{nc}_\mathrm{z} &= \frac{b_c^2(\mathrm{pc}_\mathrm{z} - h_c)}{length} \end{aligned} \tag{32}$$

### *Reflected Direction*

Once the intersection point of the ray with the hyperbola is known the reflected direction vector can be calculated. The arrival direction vector **dd**. Referring to Equation 20 the arrival direction vector $\mathbf{V_1}$ will be **dd**. **dd** is already a unit vector so $|\mathbf{V_1}| = 1.0$. The departure vector $\mathbf{V_2}$ will be **dc**. So using the normal vector to a hyperbola shown in Equation 32 **dc** is

$$\begin{aligned}
\mathrm{dc}_x &= \mathrm{dd}_x + 2.0 * (\mathbf{dd} \bullet \mathbf{nc})\mathrm{nc}_x \\
\mathrm{dc}_y &= \mathrm{dd}_y + 2.0 * (\mathbf{dd} \bullet \mathbf{nc})\mathrm{nc}_y \\
\mathrm{dc}_z &= \mathrm{dd}_z + 2.0 * (\mathbf{dd} \bullet \mathbf{nc})\mathrm{nc}_z \qquad [29] \\
&\textit{where} \\
\mathbf{d} \bullet \mathbf{nc} &= \mathrm{d}_x\mathrm{nc}_x + \mathrm{d}_y\mathrm{nc}_y + \mathrm{d}_z\mathrm{nc}_z
\end{aligned}$$

***Source* – (4)**

First the point where the ray intersects the plane of source, **ps**, must be determined. In the SCT file the user supplies the location of the center of the source (src.ctr.x, src.ctr.y, src.ctr.z) and the radius of the source src.r.

By definition $$\mathrm{ps}_z = \mathit{src.ctr.z} = \mathrm{pc}_z + \mathit{lcd} * \mathrm{dc}_z \qquad [30a]$$

Therefore $$\mathit{lcd} = \frac{\mathit{src.ctr.z} - \mathrm{pc}_z}{\mathrm{dc}_z} \qquad [30b]$$

and $$\begin{aligned} \mathrm{ps}_x &= \mathrm{pc}_x + \mathit{lcd} * \mathrm{dc}_x \\ \mathrm{ps}_y &= \mathrm{pc}_y + \mathit{lcd} * \mathrm{dc}_y \end{aligned} \qquad [30c]$$

If the distance from **ps** to the center of the source is less that src.r then the ray is has actually hit the source.

So if $$(\mathit{src.ctr.x} - \mathrm{ps}_x)^2 + (\mathit{src.ctr.y} - \mathrm{ps}_y)^2 + (\mathit{src.ctr.z} - \mathrm{ps}_z)^2 \le \mathit{src.r} \qquad [31]$$

then the ray has hit the source.

If the ray hits the source then the intersection of the ray with the object, front and back face, will be calculated, if not then the calculations move on to the next ray.

***Front and Back Face of Object* – (5) and (6)**

The intersection of the ray with the front face of the object, **pf**, is the point where the ray from mirror C first hits the object while heading towards the source. The point where the ray leaves the object is **pb**. By definition the object must be in the form of a solid rectangle. In the model at this time the object is not rotating, only translating along the z-axis, which means the surface planes of the object are all parallel to the coordinate system axes; this will make finding **pf** a little easier.

In the SCT files the user defines the object with the location of the center (obj.ctr.x, obj.ctr.y, obj.ctr.z), the size of the voxels (obj.sz.x, obj.sz.y, obj.sz.z), and the number of voxels (obj.ngrid.x, obj.ngrid.y, obj.ngrid.z). Also parameters describing the translation of the object along the z-axis in the SCT file. ops.sctr gives the location of the object center on the z-axis; obj.ctr.z is relative to ops.sctr. The other translation parameters are ops.dctr, which is the step size, and ops.nctr, which is the number of steps. From these parameters the location of surface planes of the object can be calculated.

$$\begin{aligned}
min_x &= obj.ctr.x - obj.sz.x * \frac{obj.ngrid.x}{2} \\
max_x &= obj.ctr.x + obj.sz.x * \frac{obj.ngrid.x}{2} \\
min_y &= obj.ctr.y - obj.sz.y * \frac{obj.ngrid.y}{2} \\
max_y &= obj.ctr.y + obj.sz.y * \frac{obj.ngrid.y}{2} \\
min_z &= obj.ctr.x + ops.sctr.z + ops.dctr * n - obj.sz.x * \frac{obj.ngrid.z}{2} \\
max_z &= obj.ctr.z + ops.sctr.z + ops.dctr * n + obj.sz.z * \frac{obj.ngrid.z}{2}
\end{aligned} \tag*{[32]}$$

where $n$ = 0, 1, 2,   ops.nctr-1

The ray is defined by the location vector **pc** and the direction vector **dc**. The distance from point **pc** to each plane defined by the Equations 32 needs to be calculated.

$$\begin{aligned}
lcd_{xmin} &= \frac{\left(min_x - \mathrm{pc_x}\right)}{\mathrm{dc_x}} \\
lcd_{xmax} &= \frac{\left(max_x - \mathrm{pc_x}\right)}{\mathrm{dc_x}} \\
lcd_{ymin} &= \frac{\left(min_y - \mathrm{pc_y}\right)}{\mathrm{dc_y}} \\
lcd_{ymax} &= \frac{\left(max_y - \mathrm{pc_y}\right)}{\mathrm{dc_y}} \\
lcd_{zmin} &= \frac{\left(min_z - \mathrm{pc_z}\right)}{\mathrm{dc_z}} \\
lcd_{zmaz} &= \frac{\left(max - \mathrm{pc_z}\right)}{\mathrm{dc_z}}
\end{aligned} \tag*{[33]}$$

One of these distances will defined the distance to the front face and one to the back face. If $\mathrm{dc}_{\mathbf{i}}$ is 0.0 the $lcd_{\mathbf{imin}}$ and $lcd_{\mathbf{imax}}$ will be infinity; one will be negative infinity and one postive infinity. It turns out that if the *lcd's* are put in monotonically increase order the third value will indicate the front face and the fourth value will indicate the back face. With the *lcd* the vectors **pf** and **pb** can be determined.

$$\begin{aligned}
\mathbf{pf} &= \mathbf{pc} + lcd_f * \mathbf{dc} \\
\mathbf{pb} &= \mathbf{pc} + lcd_b * \mathbf{dc}
\end{aligned} \tag*{[34]}$$

With these vectors and the direction vector **dc** the attenuation through the object can be calculated.

### Attenutation

As mentioned earlier attenuation is calculated by multiplying the linear attenuation coefficient, and the length of the ray in each voxel the ray passes through, and then summing all these terms together. The voxels the ray passes through are identified by indices *i*, *j*, and *k*. The 2D case is shown in Figure 21. The length in each voxel is identified by it's order along the line, ie. $\mathbf{l_0}$, $\mathbf{l_1}$, $\mathbf{l_2}$, ….

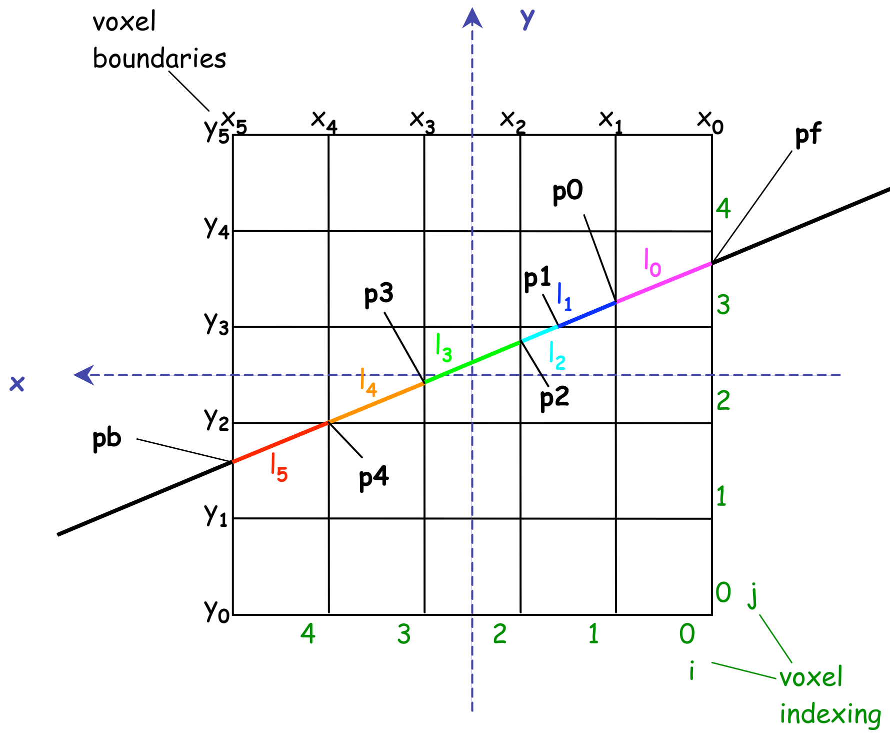


Figure 21 – Attenuation Parameters

The linear attenuation coefficients are input from the object data file and are stored in order starting from the smallest voxel on the x, y, and z axes; the x-axis increases fastest, then the y-axis and last of all the z-axis. The linear attenuation coefficient for a voxel can be accessed by knowing the voxel index *ijk*; therefore $\mu_{ijk}$.

The first step is to determine the voxel location (i,j,k) of **pf**.

$$i = \mathrm{int}\left[\frac{\left(\mathrm{pf}_x + \left(\dfrac{obj.sz.x * obj.ngrid.x}{2}\right) - obj.ctr.x\right)}{obj.sz.x}\right] \qquad [35a]$$

$$j = \mathrm{int}\left[\frac{\left(\mathrm{pf}_{\mathrm{y}} + \left(\frac{obj.sz.y * obj.ngrid.y}{2}\right) - obj.ctr.y\right)}{obj.sz.y}\right] \qquad [35b]$$

$$k = \mathrm{int}\left[\frac{\left(\mathrm{pf}_{\mathrm{z}} + \left(\frac{obj.sz.z * obj.ngrid.z}{2}\right) - \left(obj.ctr.z + ops.sctr + n * ops.dctr\right)\right)}{obj.sz.z}\right] \qquad [35c]$$

Now the direction of the ray is needed, it will be indicated by a vector, **sd**, whose values are either 1 or –1 to indicate direction along each axis.

$$\begin{aligned}
\mathrm{sd}_{\mathrm{x}} &= -1 \quad \mathit{if}\ \mathrm{pb}_{\mathrm{x}} \le \mathrm{pf}_{\mathrm{x}} \quad \mathit{otherwise}\ \mathrm{sd}_{\mathrm{x}} = 1 \\
\mathrm{sd}_{\mathrm{y}} &= -1 \quad \mathit{if}\ \mathrm{pb}_{\mathrm{y}} \le \mathrm{pf}_{\mathrm{y}} \quad \mathit{otherwise}\ \mathrm{sd}_{\mathrm{y}} = 1 \\
\mathrm{sd}_{\mathrm{z}} &= -1 \quad \mathit{if}\ \mathrm{pb}_{\mathrm{z}} \le \mathrm{pf}_{\mathrm{z}} \quad \mathit{otherwise}\ \mathrm{sd}_{\mathrm{z}} = 1
\end{aligned} \qquad [36]$$

Then the value of the next voxel boundary, **vd**, can be determined.

$$\begin{aligned}
\mathrm{vd}_{\mathrm{x}} &= i * obj.sz.x - \left(\frac{obj.sz.x * obj.ngrid.x}{2} + obj.ctr.x\right) \\
\mathrm{vd}_{\mathrm{y}} &= i * obj.sz.y - \left(\frac{obj.sz.x * obj.ngrid.x}{2} + obj.ctr.x\right) \\
\mathrm{vd}_{\mathrm{z}} &= i * obj.sz.z - \left(\frac{obj.sz.x * obj.ngrid.x}{2} + obj.ctr.x + ops.sctr + ops.dctr * n\right)
\end{aligned} \qquad [37]$$

Now the distance from **pf** to each of these voxel boundaries is calculated.

$$\begin{aligned}
lcd_x &= \frac{\mathrm{vd}_{\mathrm{x}} - \mathrm{pf}_{\mathrm{x}}}{\mathrm{dc}_{\mathrm{x}}} \\
lcd_y &= \frac{\mathrm{vd}_{\mathrm{y}} - \mathrm{pf}_{\mathrm{y}}}{\mathrm{dc}_{\mathrm{y}}} \\
lcd_z &= \frac{\mathrm{vd}_{\mathrm{z}} - \mathrm{pf}_{\mathrm{z}}}{\mathrm{dc}_{\mathrm{z}}}
\end{aligned} \qquad [38]$$

Of course if $\mathrm{dc}_{\mathrm{i}}$ is zero the $lcd_{\mathrm{i}}$ is infinity which means the ray is parallel to that axis and will not intercept it. The first boundary that the ray hits will be the one with the smallest $lcd_{\mathrm{i}}$. Define the smallest $lcd_{\mathrm{i}}$ as *lcd;* this value is $l_0$ the length through the voxel. The *i, j, k* calculated in Equation 35 define the $\mu$, so the attenuation in this voxel can now be calculated.

Now to move on to the next voxel; the point where the ray hit the first voxel boundary, **p0**, must be calculated, this is simply the *lcd* from point **pf**.

$$\mathbf{p0} = \mathbf{pf} + lcd * \mathbf{dc} \qquad [39]$$

The next voxel location can easily be determined; whichever axis boundary point **p0** is on must have its voxel index, *i, j,* or *k,* incremented by the corresponding component of the direction vector **sd**. If more than one voxel boundary was reached, such as p4 in Figure 21, then all the indices involved will be incremented.

This procedure will be repeated until **pb** is reached.

## IMAGING STUDIES

This section will examine how a Wolter Optic creates images. The way the system works is not intuitively obvious so a number of features will be examined individually.

A number of simulations were run with the model code. These simulations are described in the next section; they will however be referenced as examples of the features being examined in this section.

The code takes a long time to run and there are memory restrictions so it was not possible to run simulations as large as desired with the maximum resolution, so two different types of tests were done. One type had an image with large physical dimensions and large voxel and pixel sizes, in order to keep the total number of voxels and pixels small. These tests were used to examine the effective field of view. The other type of simulations had small physical size with small resolution in order to study in more detail some of the image effects.

### Magnification

The magnification of the system is defined in Equation 3a as $\boldsymbol{M} = \sin\theta_{\mathbf{cb}}/\sin\theta_{\mathbf{df}}$ which is approximately equal to $\boldsymbol{M} = d_{\mathbf{2}}/d_{\mathbf{1}}$ where $d_{\mathbf{1}}$ and $d_{\mathbf{2}}$ are shown in Figure 22. A point which is a distance $\boldsymbol{x}$ from the z-axis on the focal plane will be at distance $\boldsymbol{M*x}$ from the z-axis on the image plane, **on the opposite** side of the z-axis from the original point.

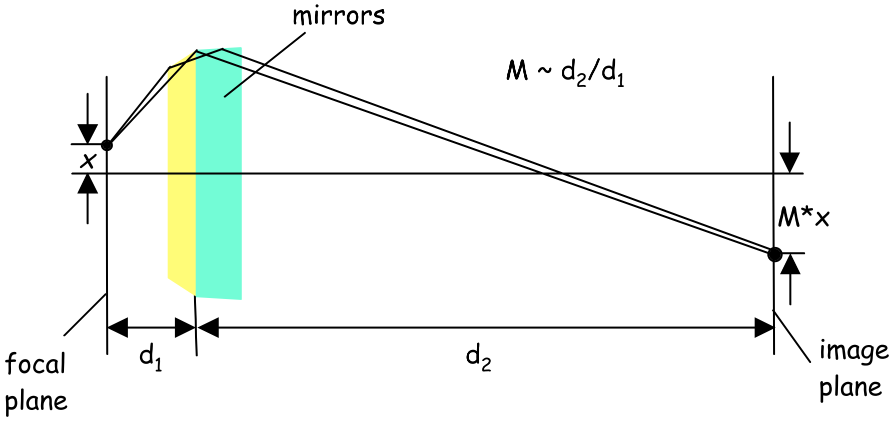


Figure 22 – Image Magnification

This effect is shown in all the simulations because in the pixel sizes are always 12x larger than the voxel sizes. Since all the images on the focal plane appear the same, only reversed, on the image plane the magnification works as expected.

### Image Formation/Reversal

As mentioned in the magnification section the image formed is of the object reversed.  Figure 23 shows how this occurs with points from both sides of the z-axis.

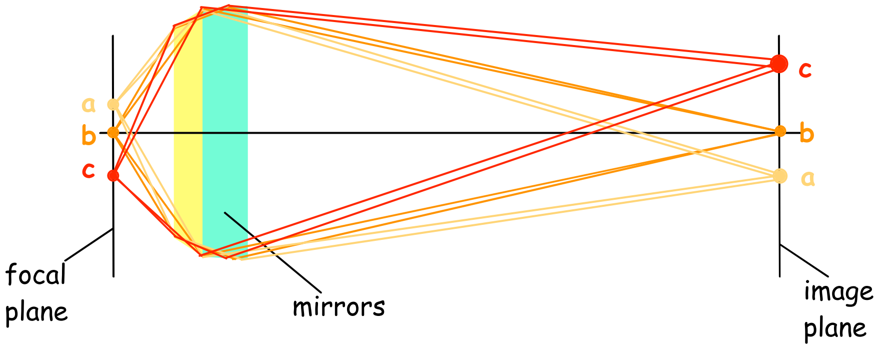


Figure 23 – Formation of Image Reversal

Since the mirrors are axis-symmetric the image is a complete reversal, Figure 24 tries to show this clearly.

focal plane

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 12 | | | 5 |
| 11 | | | 6 |
| 10 | 9 | 8 | 7 |

image plane

| | | | |
|---|---|---|---|
| 7 | 8 | 9 | 10 |
| 6 | | | 11 |
| 5 | | | 12 |
| 4 | 3 | 2 | 1 |

Figure 24 – Image Reversal

Image reversal is seen clearly in Simulations 1-7.

### Off-Axis Blurring

Off-axis blurring refers to an affect that occurs to points on the focal plane, but off the z-axis. The x-rays passing through points on the focal plane but off-axis will strike at different angles and positions all around the mirror and eventually, as the distance from the z-axis increases the image on the image plane will blur. This is shown in Figure 25.  The field of view of the system is the size of the object that can be place at the focal plane and remain in focus on the image plane. This image blurring will of course also occur when the object is not on the focal plane, but it will be in combination with other affects.  Off-axis blurring is seen best in Simulations 2 and 3.

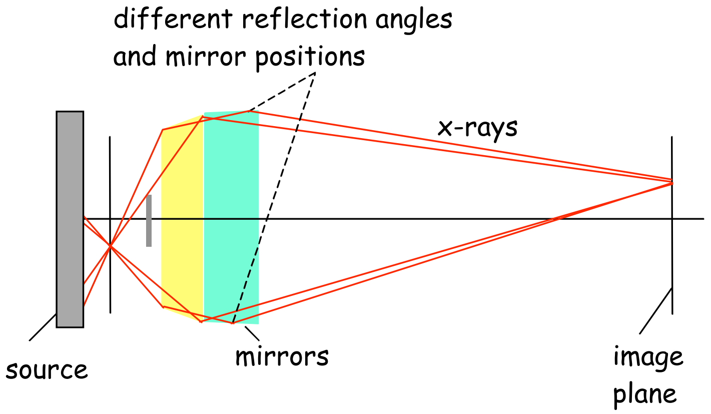


Figure 25 – Off-Axis Blurring

### Off-Axis Loss of Image

As a point gets too far off the z-axis some of the x-rays going from the source through the point and on to the mirror will hit the mirror at too large of an angle to reflect. This is shown in Figure 26. This will cause loss of image strength on the image plane. The blurred image will also look less symmetric than a blurred image without this effect. This loss of image is seen at the farthest off-axis points in Simulations 2 and 3.

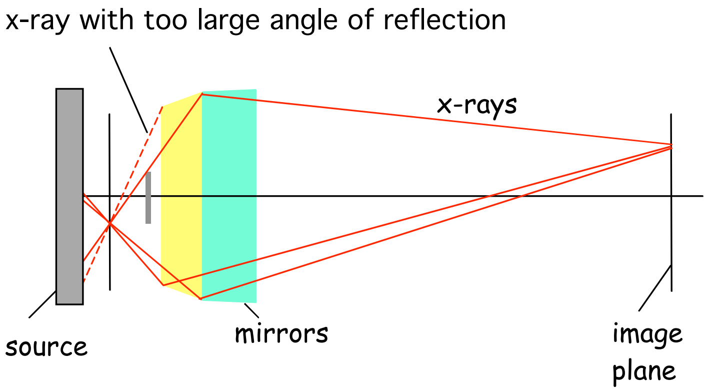


Figure 26 – Off-Axis Loss of Image

### Off-Focal Plane Image/Resolution

An X-ray image of an object is essentially a shadow of the object; a 2D image of a 3D object. In a standard X-ray each pixel on the projection is the end point of one x-ray; it represents one path through the object. In a Wolter system, because of the axis-symmetric mirrors many x-ray paths will end at any given pixel of the projection. To see this effect an interesting simulation was setup. Instead of a forward simulation of an object the simulation traced rays backward from one pixel through the object. As explained in the Code Description section, the user can selected the number of ray paths from a pixel. 24 rotational angles were selected and only one z-axis depth. Each of the 24 rotational rays was given a different color and each time the ray intersected an object voxel that color was added to the voxel. Figure 27 shows the result. It can be seen that the beams form circles on the off focal plane planes. In a forward simulation the points on these circles are the voxels that will be collected and summed and passed onto the image pixel. So the shadow image that the projection represents will definitely be different than a standard CT.

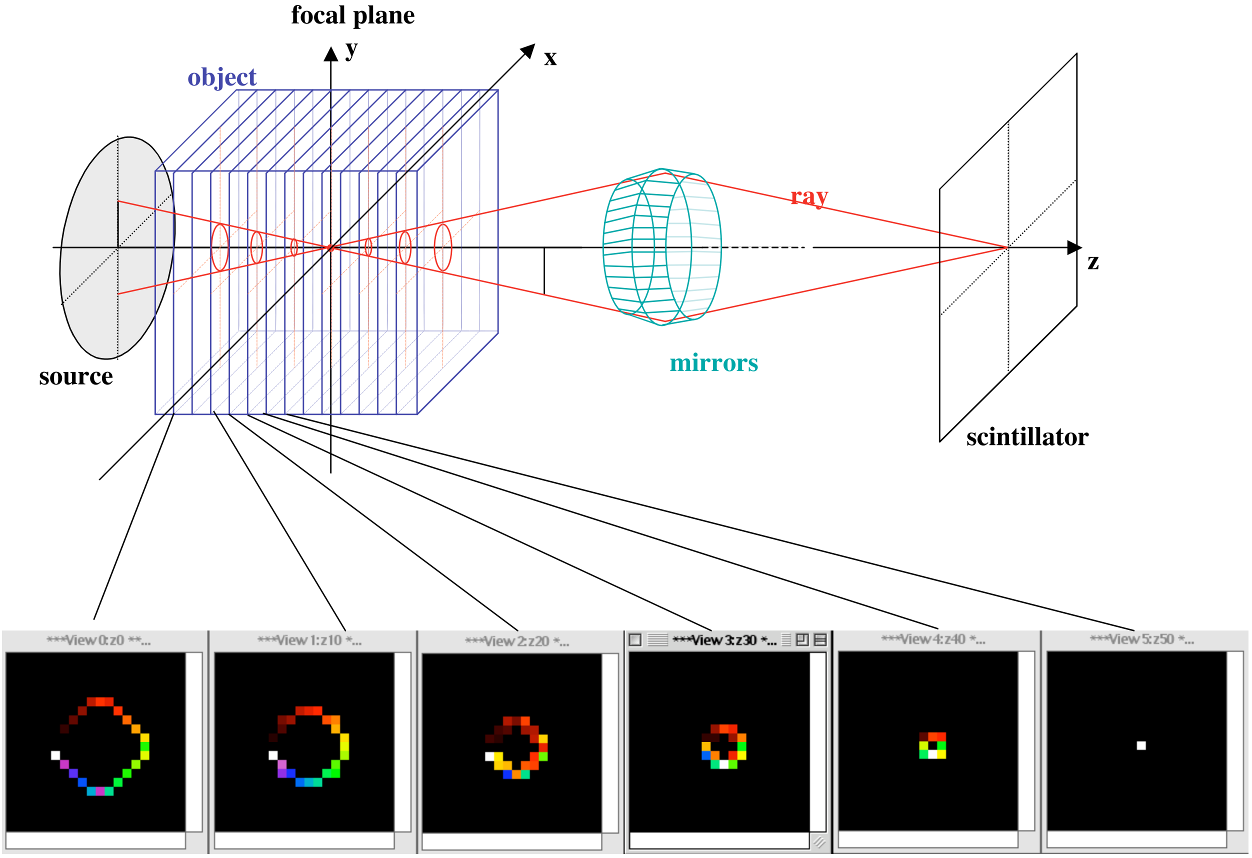


Figure 27 -

The point of a Wolter Optic is to magnify the object on the focal plane; which this system will do, however the magnified images from all the off-focus planes will be added to the magnified plane of interest. This non-uniqueness of projection data brings up an interesting question of whether it will be possible to develop a reconstruction routine for Wolter Optic data.

Off-focal plane resolution can be seen in Simulations 1-8.

## RECOMMENDATIONS

After examining the simulations it can be seen that within the field-of-view of the design the system magnifies and focuses images on the focal plane to within the desired resolution (0.0005 mm). However the z-axis resolution is much larger; almost 10 times larger (0.005 mm). It would be interesting to see if a design could be developed to improve the resolution along the z-axis.

Before further development work is done it would be useful to continue the code development and simulation studies outlined in the Code Development section. Even though the model code only examines a "perfect" form of the system, if it shows the images are not useful then images from a real system are unlikely to be useful.

## SIMULATIONS

All the simulations presented in this section use the following test setup; the z-axis is the instrument axis of the Wolter Microscope, the focal, or object, plane is at z = 0, the y-axis extends up from z = 0, and the x-axis extends into the paper.  The simulations involve stepping the object along the z-axis as shown by Figures  28a, 28b and 28c.

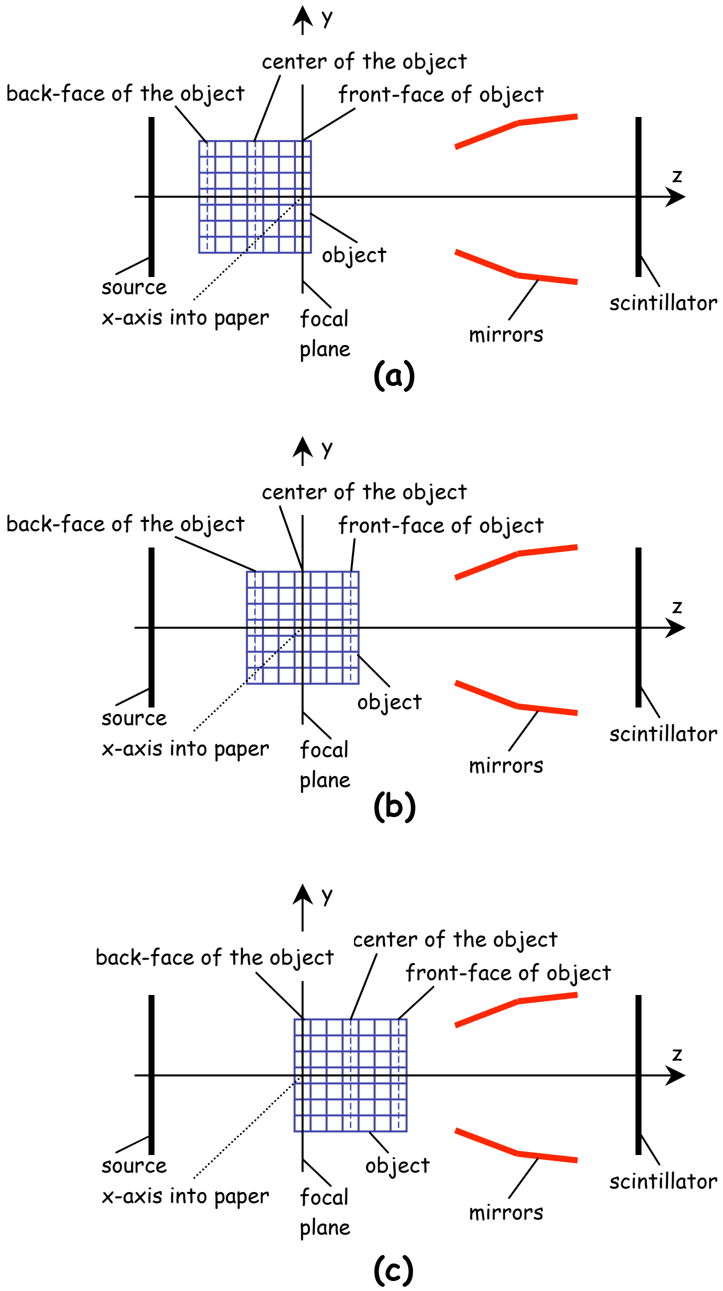


Figure 28 – Motion of Object During Simulations; a) back face aligned with focal plane, b) center aligned with focal plane, c) front face aligned with focal plane

## Simulation 1

### *Test*

This is a test with a single voxel. A series of images is simulated with the voxel starting on the focal plane and moving farther and farther off the focal plane. The voxel is placed off-axis to examine the asymmetric affect of the imaging.

The result is displayed as a small area around the voxel.

### *Observations*

- The image is magnified
- The image is reversed.
- At a distance from the z-axis of 0.05, the image is in focus when on the focal plane.
- The voxel stays in focus until 0.006 (3 slice steps) off the focal plane.
- The voxel is off-axis and the asymmetry of the mirror paths can be seen in the shape of the image as the object moves off the focal plane.

### *Object*

Each voxel is 0.002 mm cube.

This object in this test is 51 x 51 voxels or

0.102 mm x 0.102 mm

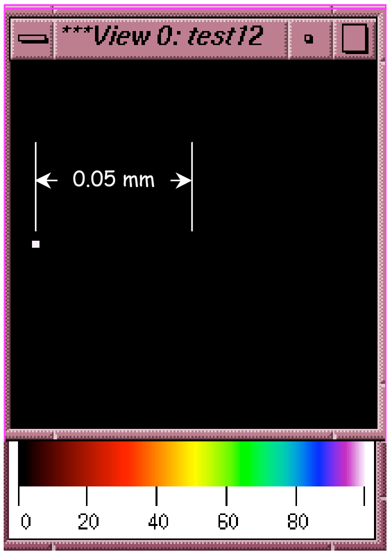
all zeros except 100 at (3,25)

### *Scintillator*

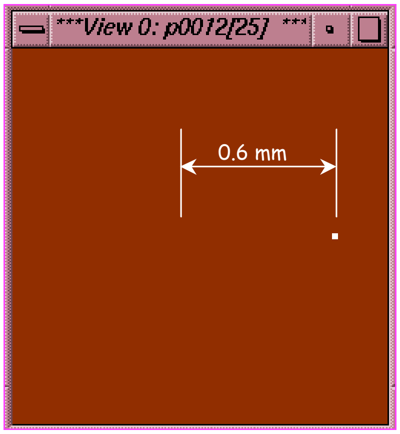
61 x 61 pixels; pixel = 0.024mm

1.464 mm x 1.464 mm

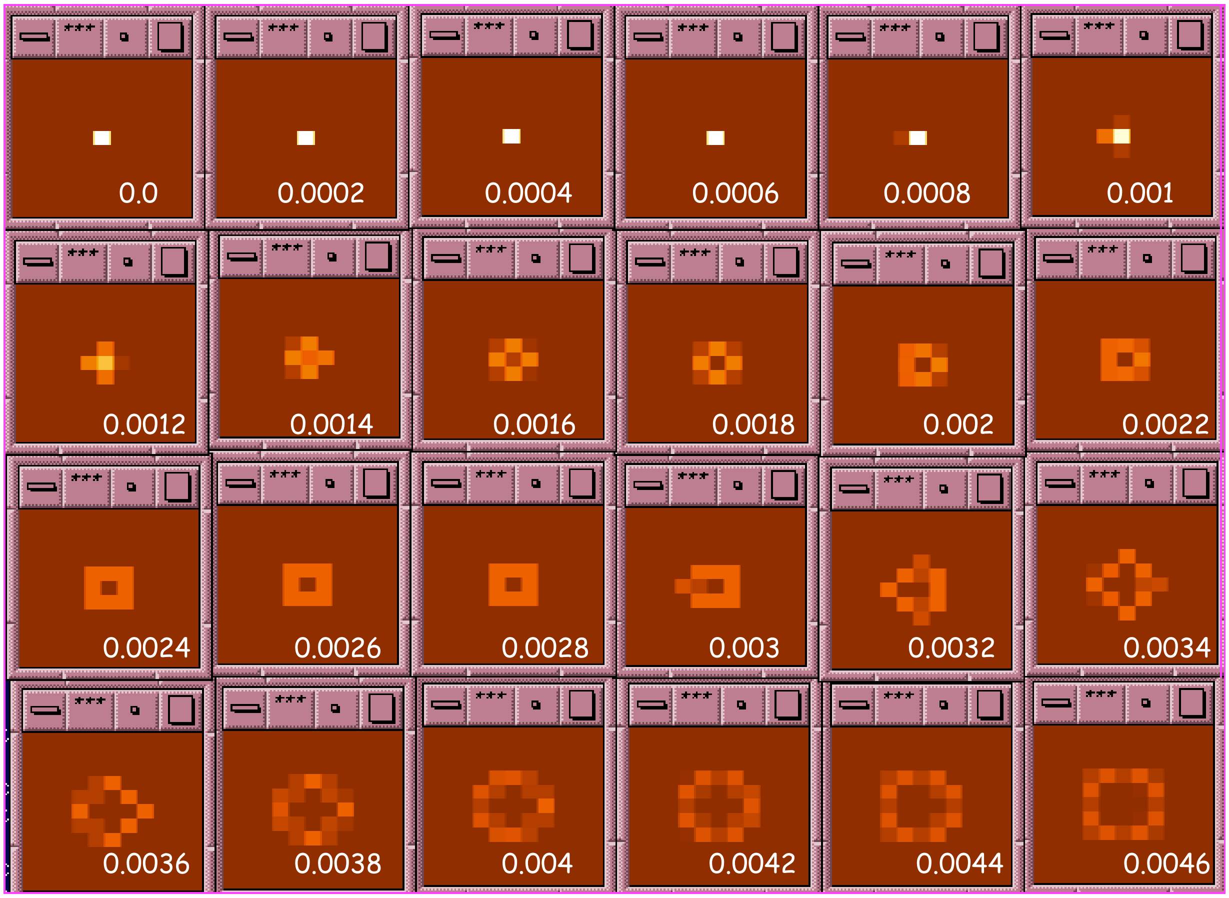
***Object***



***Over view of first projection***



***Resulting Projections*** – Shown is the distance (mm) of the object center from the focal plane



**Simulation 2**

***Test***

This test has a series of single voxels separated by space extending out 0.5 mm. This allows the off-axis blurring and loss of image to be observed. The test is run by placing the object centered on the focal plane; the object is then stepped along the z-axis until the back face is on the focal plane. The projections shown are 0-5, and only the strip around the zero voxels are shown.

***Observations***

- The image is magnified.
- The image is reversed, this is true though it can not be seen in the images shown here.
- When the test voxels are on the focal plane they stay in focus until about 0.2 mm off-axis.
- There is asymmetry in the voxels off-axis.
- The voxels near the z-axis stay in focus until the object plane is about 0.006 mm off the focal plane.

***Object***

Each voxel is 0.002 mm cube.

This object in this test is 501 x 501 x 10 voxels or

1.002 mm x 1.002 mm x 0.02 mm

background all value 100

a row of 25 zeros along the x=0 axis, single voxels, 10 voxels (0.02 mm) apart

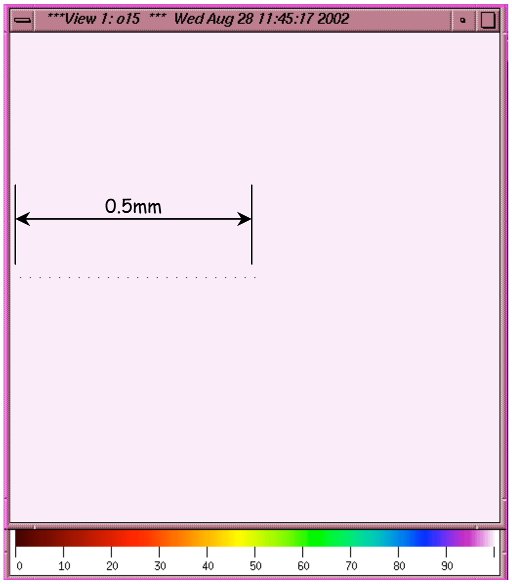
total length 0.50 mm

***Scintillator***

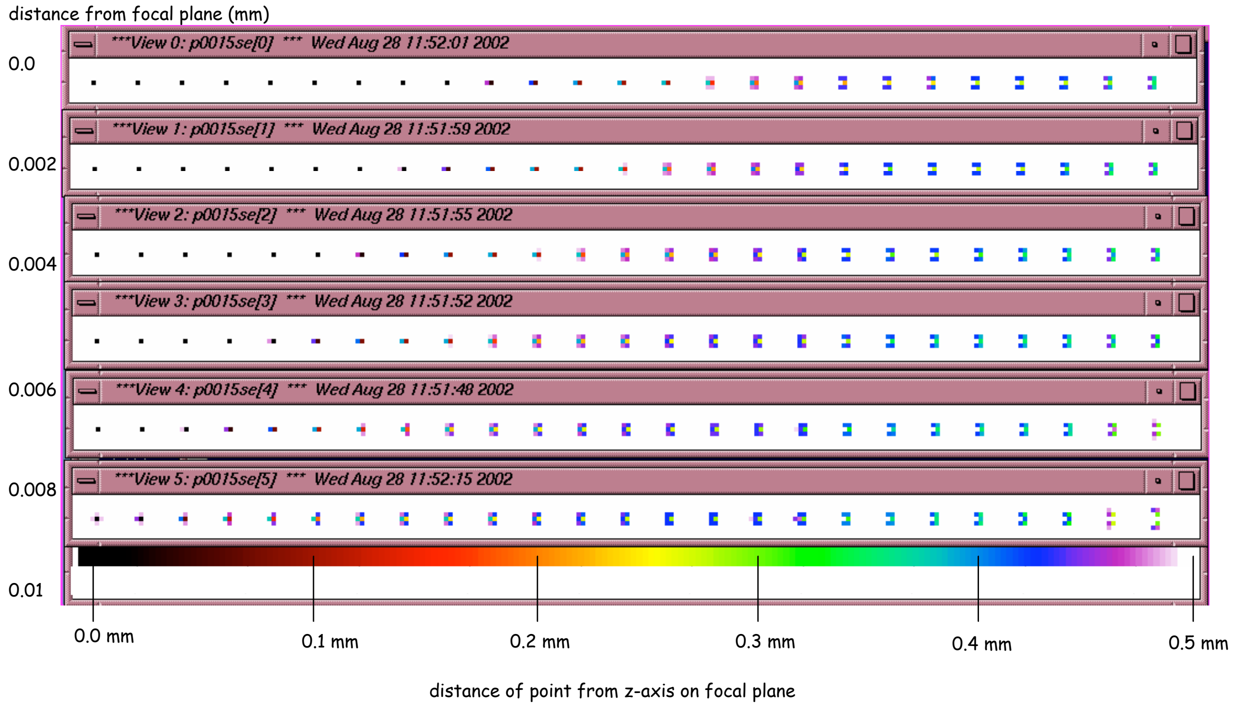
501 x 501 pixels; pixel = 0.024 mm

0.12024 mm x 0.12024 mm

## *Object*



## *Resulting Projections*


distance from focal plane (mm)
0.0
0.002
0.004
0.006
0.008
0.01
***View 0: p0015se[0] *** Wed Aug 28 11:52:01 2002
***View 1: p0015se[1] *** Wed Aug 28 11:51:59 2002
***View 2: p0015se[2] *** Wed Aug 28 11:51:55 2002
***View 3: p0015se[3] *** Wed Aug 28 11:51:52 2002
***View 4: p0015se[4] *** Wed Aug 28 11:51:48 2002
***View 5: p0015se[5] *** Wed Aug 28 11:52:15 2002
0.0 mm
0.1 mm
0.2 mm
0.3 mm
0.4 mm
0.5 mm
distance of point from z-axis on focal plane

## Simulation 3

### *Test*

This test has a series of single voxels separated by space extending out 0.5 mm. This allows the off-axis blurring and loss of image to be observed. The test is run by placing the object centered on the focal plane; the object is then stepped along the z-axis until the back face is on the focal plane. The projections shown are 0-5, and only the strip around the zero voxels are shown.

### *Observations*

- The image is magnified.
- The image is reversed, this is true though it can not be seen in the images shown here.
- When the test voxels are on the focal plane they stay in focus until about 0.2 mm off-axis.
- There is asymmetry in the voxels off-axis.
- The voxels near the z-axis stay in focus until the object plane is about 0.006 mm off the focal plane.

### *Object*

Each voxel is 0.002 mm cube.

This object in this test is 501 x 501 x 10 voxels or

1.002 mm x 1.002 mm x 0.02 mm

background all value 100

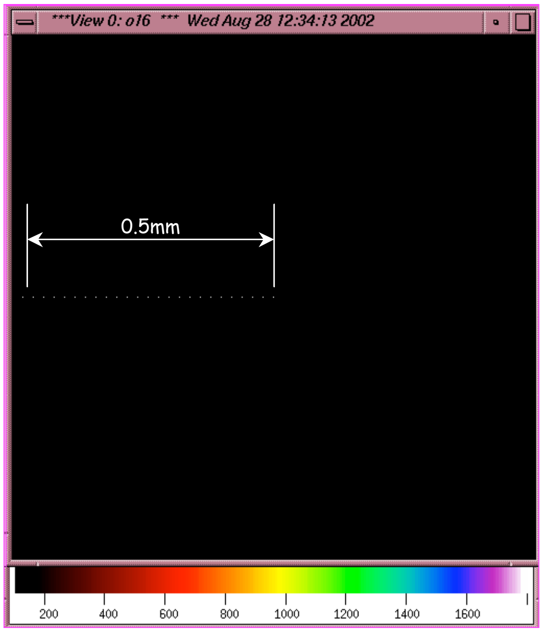
a row of 25 points of value 1800 along the x=0 axis, single voxels, 10 voxels (0.02 mm) apart; total length 0.50 mm

### *Scintillator*

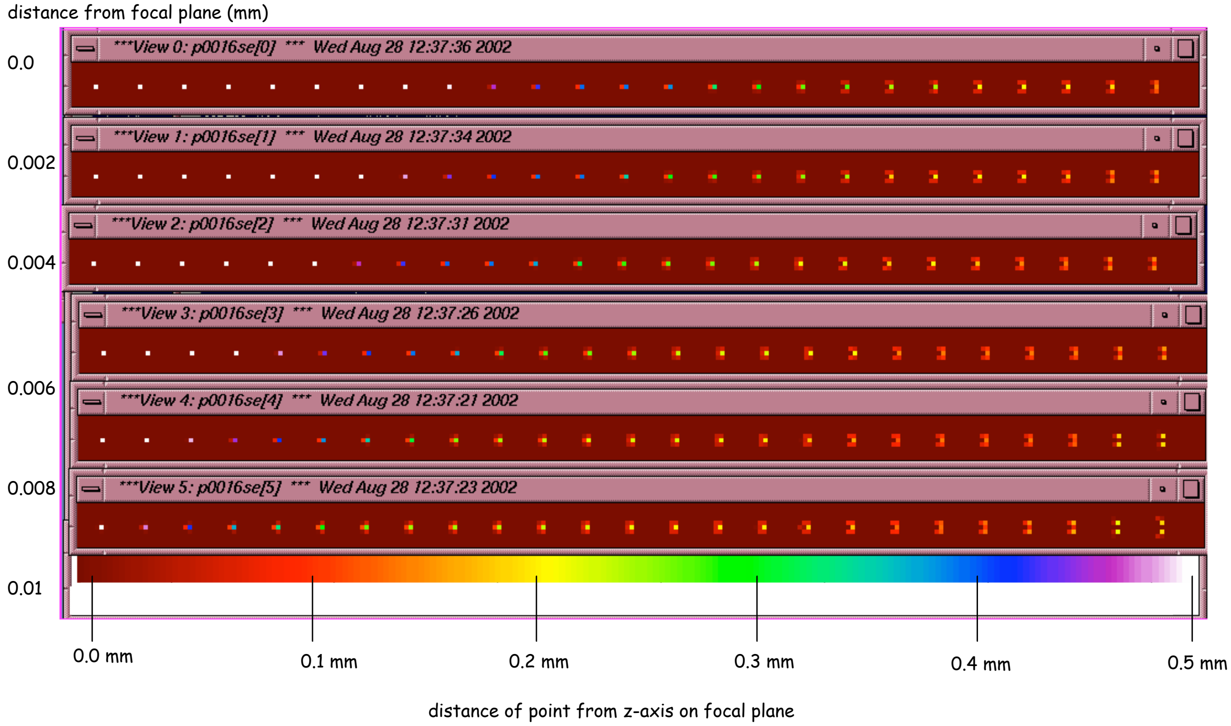
501 x 501 pixels; pixel = 0.024 mm

0.12024 mm x 0.12024 mm

### *Object*



### *Resulting Projections*


distance from focal plane (mm)
0.0
0.002
0.004
0.006
0.008
0.01
***View 0: p0016se[0] *** Wed Aug 28 12:37:36 2002
***View 1: p0016se[1] *** Wed Aug 28 12:37:34 2002
***View 2: p0016se[2] *** Wed Aug 28 12:37:31 2002
***View 3: p0016se[3] *** Wed Aug 28 12:37:26 2002
***View 4: p0016se[4] *** Wed Aug 28 12:37:21 2002
***View 5: p0016se[5] *** Wed Aug 28 12:37:23 2002
0.0 mm
0.1 mm
0.2 mm
0.3 mm
0.4 mm
0.5 mm
distance of point from z-axis on focal plane

## Simulation 4

### *Test*

The test is to observe the focusing. This object is thicker and the "flaw" is in the center.

The object is placed so the front face is on the focal plane, the object is then stepped along the z-axis until the back face is on the focal plane. Each step is the size of a slice. Shown here are projections 25-39, or from when the object is center on the focal plane out 15 steps.

### *Observations*

- The object is magnified.
- The object is reversed.
- When the "flaw" is on the focal plane it can be seen clearly.
- The off-focal plane image resolution is the same until 0.006-0.008 mm off the focal plane.

### *Object*

Each voxel is 0.002mm cube.

This object in this test is 51 x 51 x 51 voxels or

0.102mm x 0.102mm x 0.102mm

two concentric spheres

sphere 1 – radius = 10 voxels = 0.02mm; value = 1000

sphere 2 – radius = 20 voxels = 0.04mm; value = 500

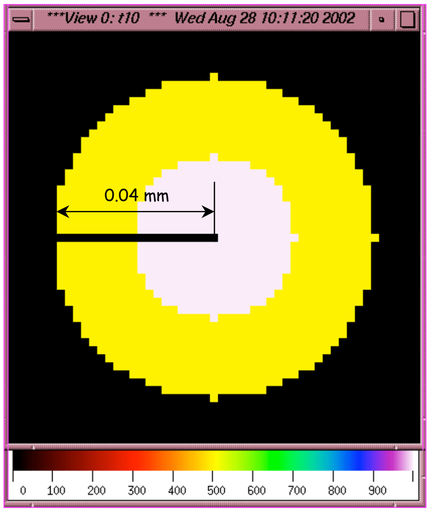
a line of value 0 is placed on the center slice of the object from (0, 25, 25) to (25, 25, 25)

### *Scintillator*

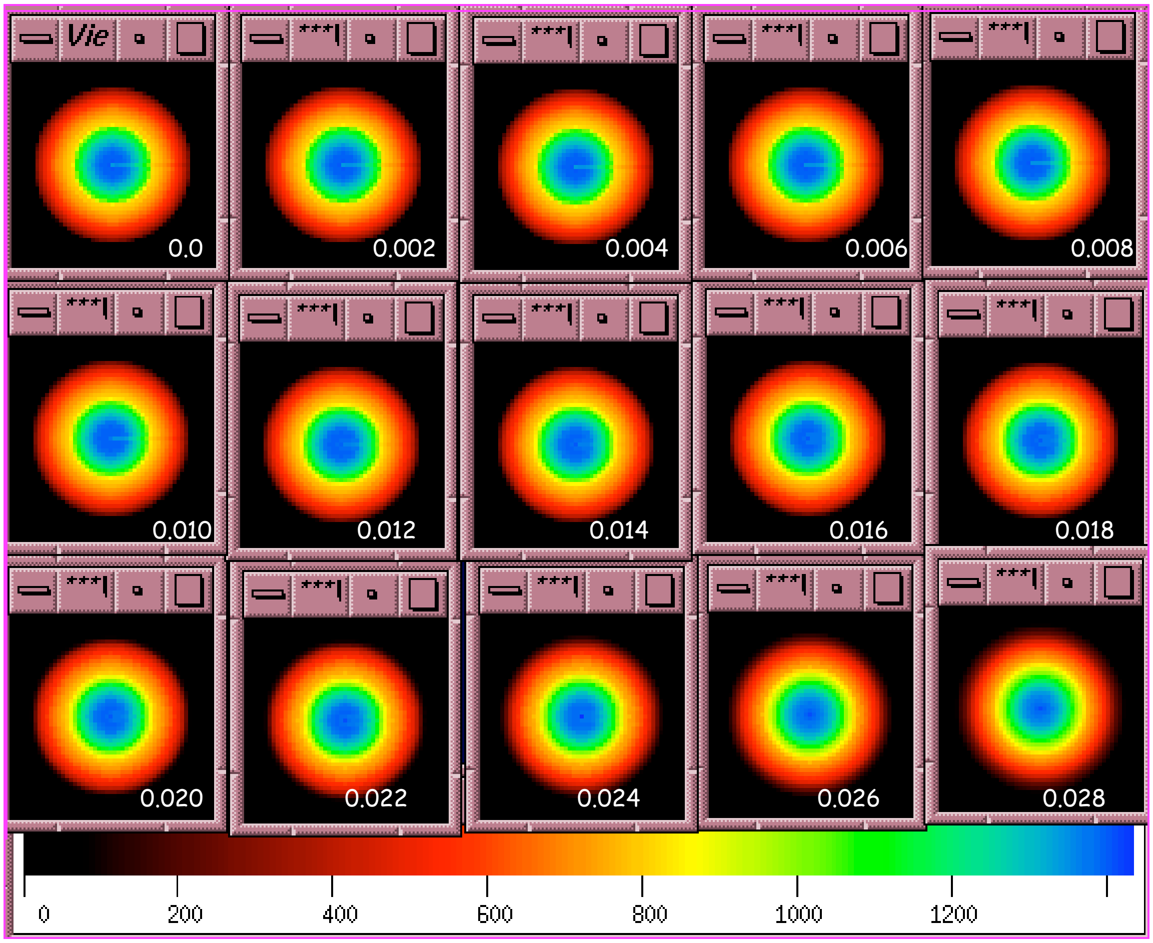
51 x 51 pixels; pixel = 0.024mm

1.224 mm x 1.224 mm

***Center slice of object***



***Resulting Projections*** **–** Shown is the distance (mm) of the object center from the focal plane



## Simulation 5

### *Test*

The test is to observe the focusing. This object is thicker and the "flaw" is in the center.

The object is placed so the front face is on the focal plane, the object is then stepped along the z-axis until the back face is on the focal plane. Each step is the size of a slice. Shown here are projections 25-39, or from when the object is center on the focal plane out 15 steps.

### *Observations*

- The object is magnified.
- The object is reversed.
- When the "flaw" is on the focal plane it can be seen clearly.
- The off-focal plane image resolution is the same until 0.006-0.008 mm off the focal plane.

### *Object*

Each voxel is 0.002 mm cube.

This object in this test is 51 x 51 x 51 voxels or

0.102 mm x 0.102 mm x 0.102 mm

two concentric spheres

sphere 1 – radius = 10 voxels = 0.02 mm; value = 1000

sphere 2 – radius = 20 voxels = 0.04 mm; value = 500

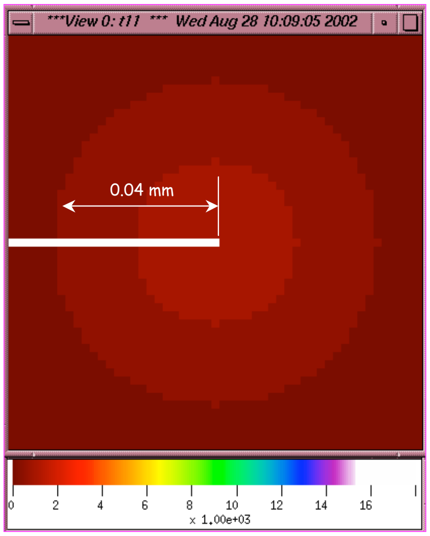
a line of value 1800 is placed on the center slice of the object from (0, 25, 25) to (25, 25, 25)

### *Scintillator*

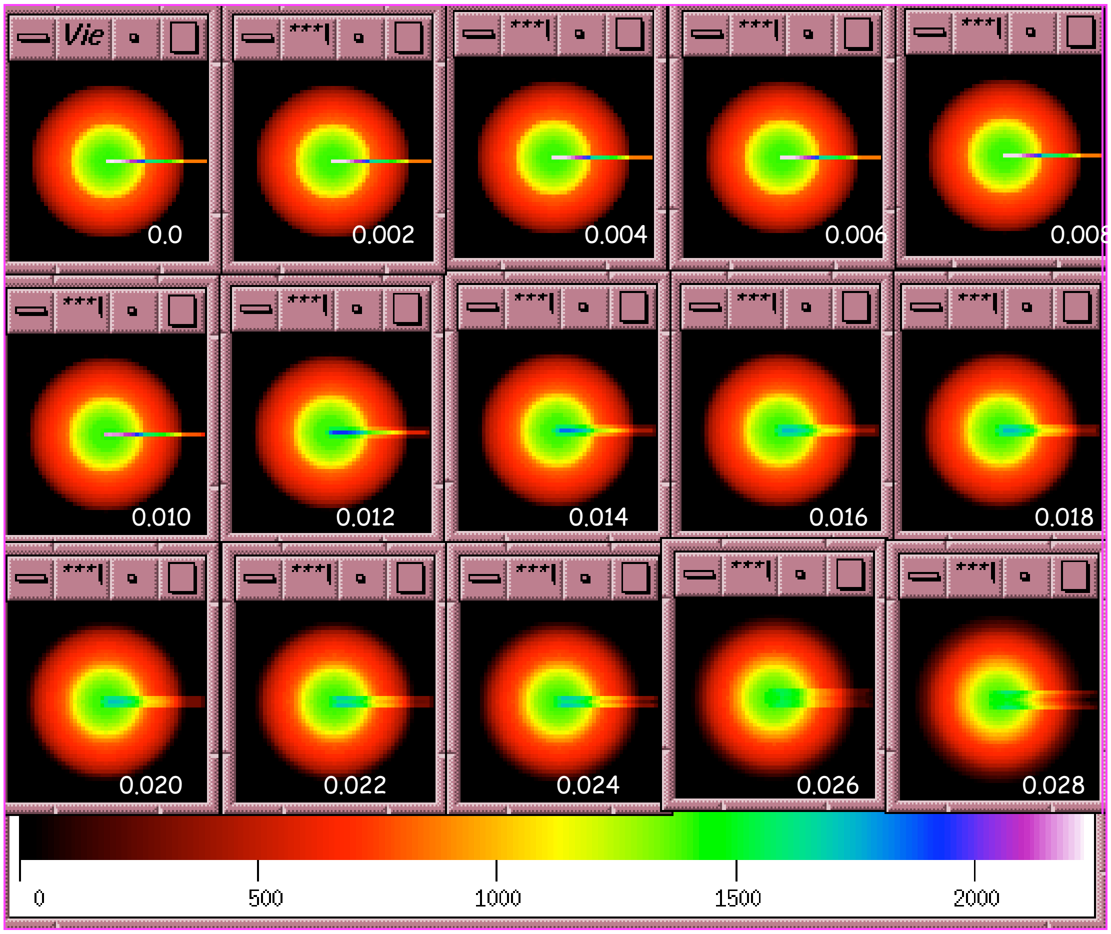
51 x 51 pixels; pixel = 0.024 mm

1.224 mm x 1.224 mm

***Center Slice of Object***



***Resulting Projections*** – Shown is the distance (mm) of the object center from the focal plane



## Simulation 6

### *Test*

The resolution of this test is smaller than the previous simulations; the resolution is at the system requirement.

The test is run by placing the object centered on the focal plane; the object is then stepped along the z-axis until the back face is on the focal plane. The projections shown are 0-10, 20, 30, 40 and 45

### *Observations*

- The object is magnified.
- The object is reversed.
- The object is not big enough to really observe off-axis blurring.
- The image stays in focus until 0.005mm off-focal plane.

### *Object*

Each voxel is 0.0005 mm cube.

This object in this test is 101 x 101 x 101 voxels or

0.0505 mm x 0.0505 mm x 0.0505 mm

one spheres

radius = 45 voxels = 0.0225 mm; value = 100

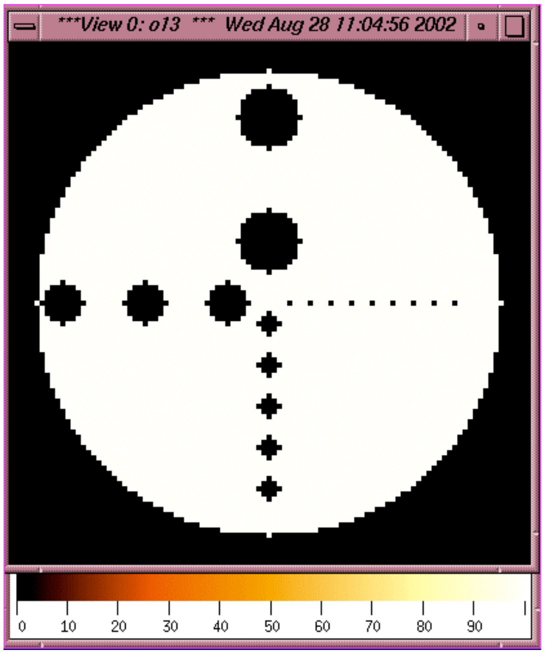
spherical voids are placed as shown in the diagram

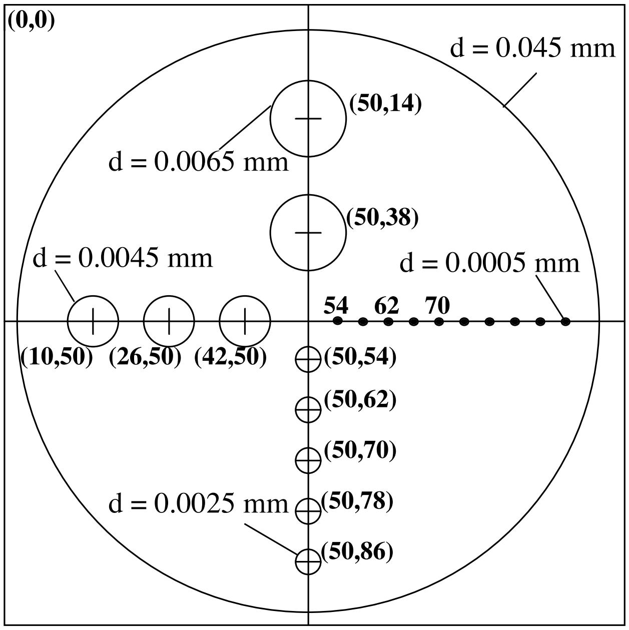
### *Scintillator*

101 x 101 pixels; pixel = 0.006 mm

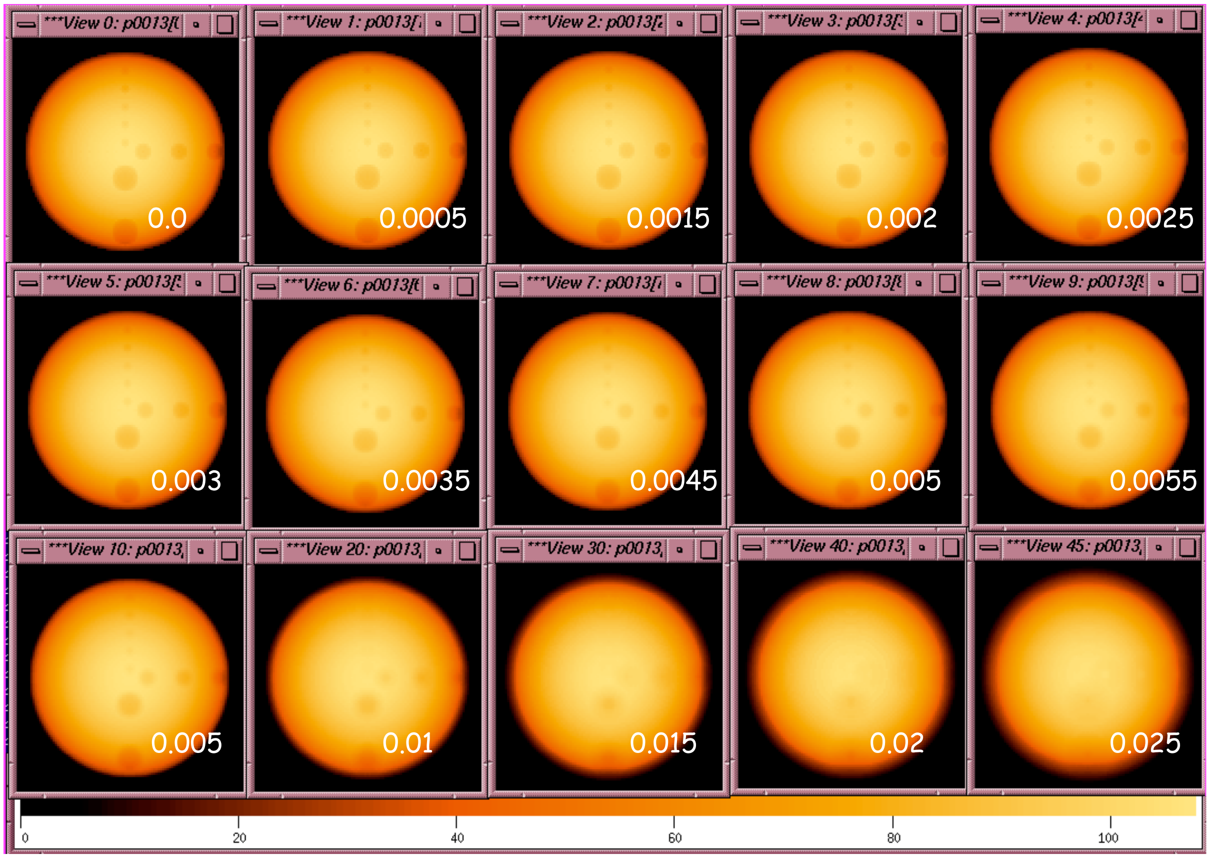
0.606 mm x 0.606 mm

***Object***





***Resulting Projections*** – Shown is the distance (mm) of the object center from the focal plane



## Simulation 7

### *Test*

The resolution of this test is smaller than the previous simulations; the resolution is at the system requirement.

The test is run by placing the object centered on the focal plane; the object is then stepped along the z-axis until the back face is on the focal plane. The projections shown are 0-10, 20, 30, 40 and 45

### *Observations*

- The object is magnified.
- The object is reversed.
- The object is not big enough to really observe off-axis blurring.
- The image stays in focus until 0.005mm off-focal plane.

### *Object*

Each voxel is 0.0005 mm cube.

This object in this test is 101 x 101 x 101 voxels or

0.0505 mm x 0.0505 mm x 0.0505 mm

one spheres

radius = 45 voxels = 0.0225 mm; value = 100

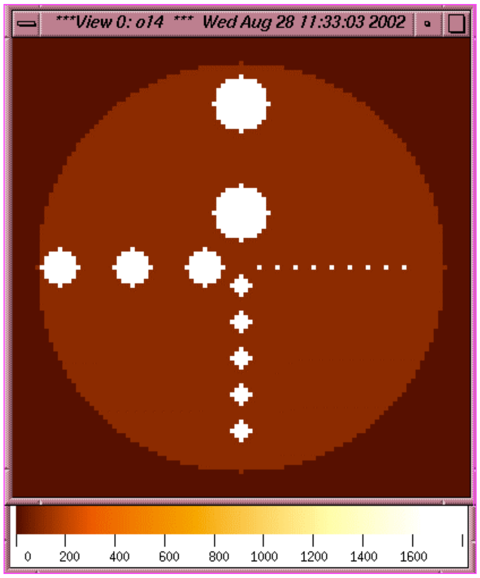
spherical inclusions of value 1800 are placed as shown in the diagram

### *Scintillator*

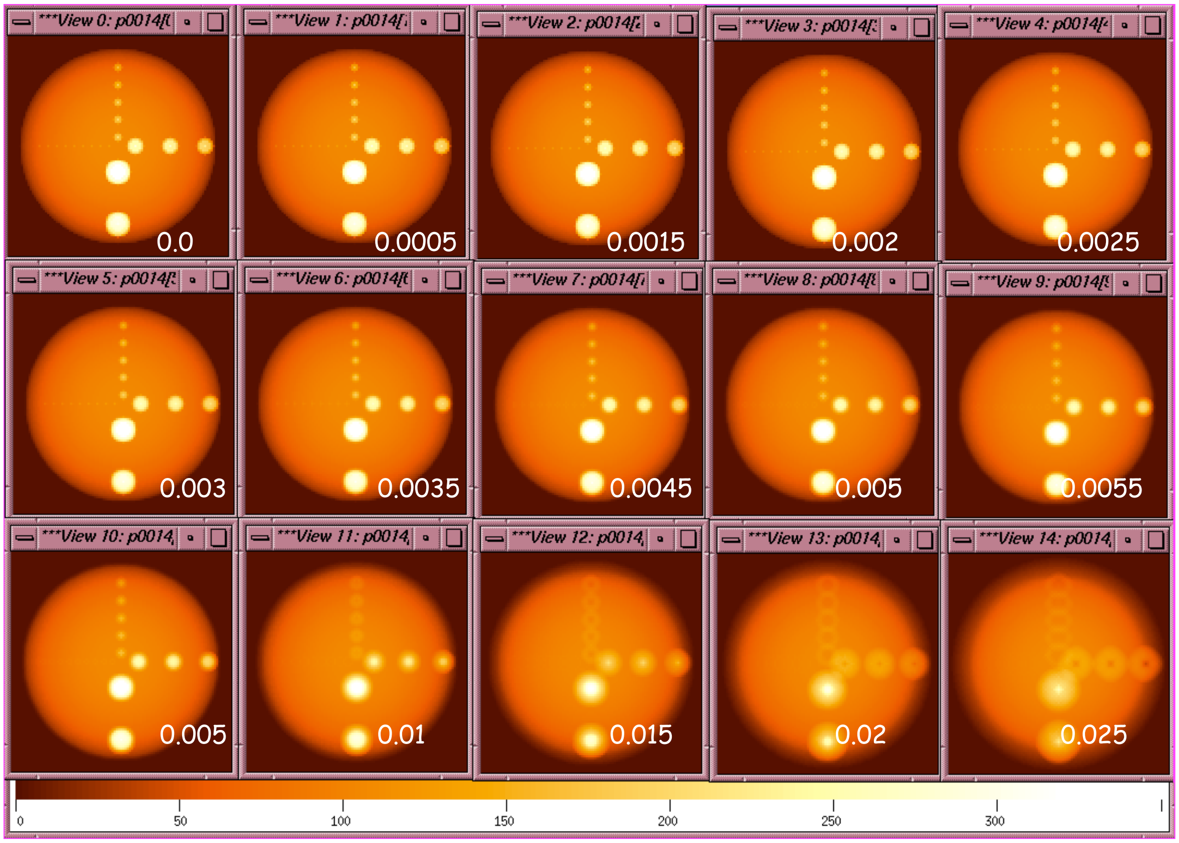
101 x 101 pixels; pixel = 0.006 mm

0.606 mm x 0.606 mm

***Object***



***Resulting Projections*** – Shown is the distance (mm) of the object center from the focal plane



## Simulation 8

### *Test*

The object in this test is the size as the design field of view (not the specified) and the resolution is the specified.

The test is run by placing the object centered on the focal plane; the object is then stepped along the z-axis until the back face is on the focal plane. Selected projections are shown.

### *Observations*

- The image is magnified.
- Because of symmetry it is hard to tell that the image is reverse.
- No off-axis blurring.
- Whether the object is off-focal plane in the positive or negative cannot be determined from the projections.

### *Object*

Each voxel is 0.0005 mm cube.

This object in this test is 251 x 251 x 251 voxels or

0.1255 mm x 0.1255 mm x 0.1255 mm

background = 0

three rectangular objects of size 50 x 50 x 201 voxels are placed inside the main object as shown, the value of each rectangle is 25.

inside each rectangular object are three sphere of radius = 5 voxels and value = 0

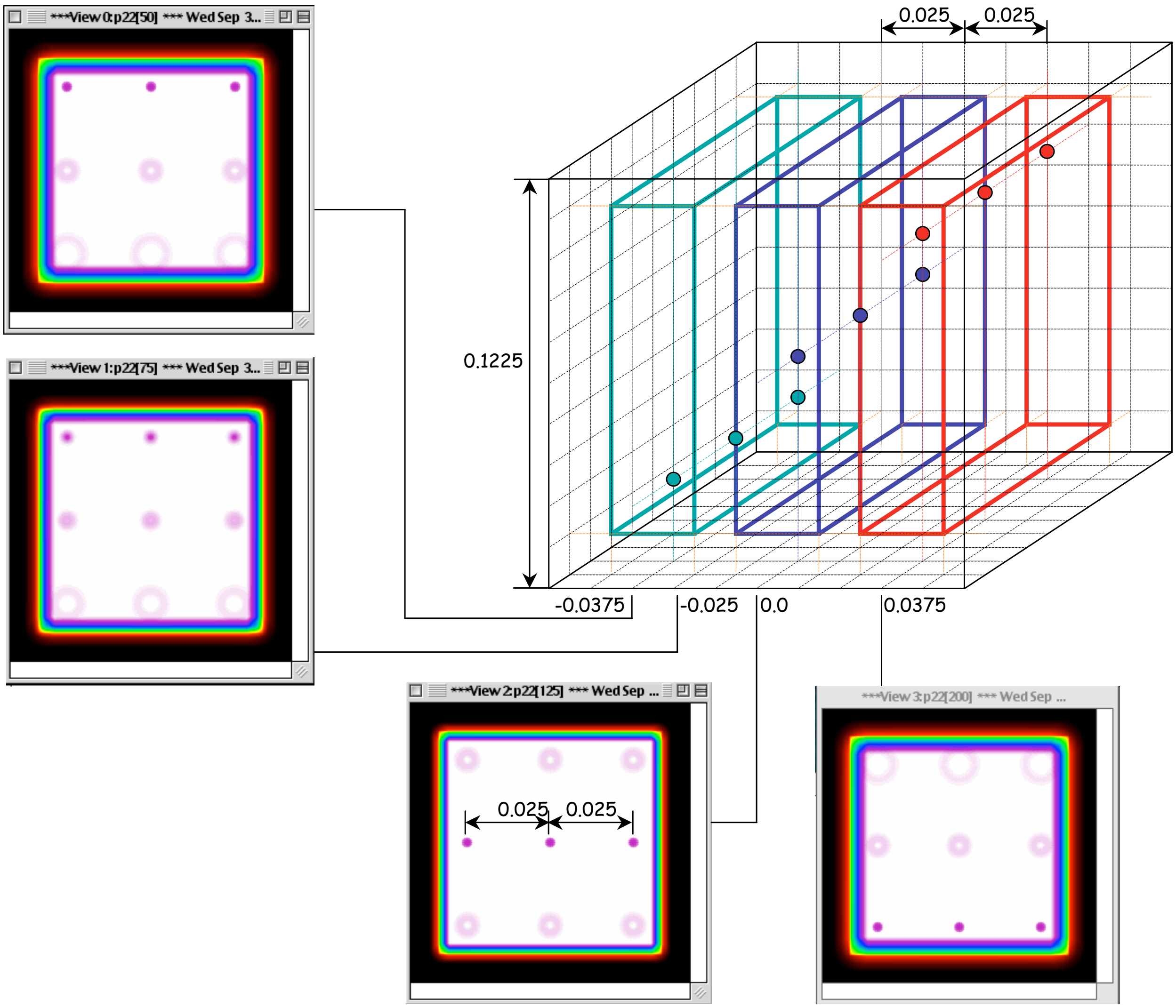
### *Scintillator*

251 x 251 pixels; pixel = 0.006 mm

0.1506 mm x 0.1506 mm

### *Object and Resulting Projections*

All dimension are in mm.


***View 0:p22[50] *** Wed Sep 3...
***View 1:p22[75] *** Wed Sep 3...
0.025
0.025
0.1225
-0.0375
-0.025
0.0
0.0375
***View 2:p22[125] *** Wed Sep ...
0.025
0.025
***View 3:p22[200] *** Wed Sep ...

# APPENDIX A – SIMULATION PARAMETER FILES

In order to run each simulation an *sct* input file is needed. The code produces an output text file with the some of the parameter values calculated during the processing. These files are included in this appendix for each of the simulations.

## Simulation 1

### SCT File

```
! Sct file for wolter microscope

-ofile           test10

-ops.power_flag        0          ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x       0.0     ! x location for center of source (mm)
-src.ctr.y       0.0     ! y location for center of source (mm)
-src.ctr.z       -0.1    ! z location for center of source (mm)
-src.r           0.1     ! radius of source (mm)

-scint.nang.a1   1       ! Number of division across D mirror
-scint.nang.a2   24      ! Number of rotation angles around D mirror
-scint.sz.x      0.024   ! Size of scintilator pixel in x direction (mm)
-scint.sz.y      0.024   ! Size of scintilator pixel in y direction (mm)
-scint.sz.z      0.024   ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x     0.0     ! x location for center of scintilator (mm)
-scint.ctr.y     0.0     ! y location for center of scintilator (mm)
-scint.ctr.z     5000.0  ! z location for center of scintilator (mm)
-scint.ngrid.x   51      ! number of pixels in x direction
-scint.ngrid.y   51      ! number of pixels in y direction

-obj.sz.x        0.002   ! Size of object voxel in x direction (mm)
-obj.sz.y        0.002   ! Size of object voxel in y direction (mm)
-obj.sz.z        0.002   ! Size of object voxel in z direction (mm)
-obj.ctr.x       0.0     ! x location for center of object (mm)
-obj.ctr.y       0.0     ! y location for center of object (mm)
-obj.ctr.z       0.0     ! z location for center of object (mm)
-obj.ngrid.x     51      ! number of voxels in x direction
-obj.ngrid.y     51      ! number of voxels in y direction
-obj.ngrid.z     51      ! number of voxels in z direction

-ops.sctr        -0.05   ! location of z center for first projection
-ops.dctr        0.002   ! step size for center along z
-ops.nctr        51      ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       0.100000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.024000000000
Scintilator - y pixel size (mm) ============       0.024000000000
Scintilator - number x pixels ==============      51.000000000000
Scintilator - number y pixels ==============      51.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============       0.050000000000
Object x voxel size (mm) ===================       0.002000000000
Object y voxel size (mm) ===================       0.002000000000
Object z voxel size (mm) ===================       0.002000000000
Object number x voxels  ====================      51.000000000000
Object number y voxels  ====================      51.000000000000
Object number z voxels  ====================      51.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

**Simulation 2**

**SCT File**

```
! Sct file for wolter microscope

-ofile            test11

-ops.power_flag        0         ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x        0.0      ! x location for center of source (mm)
-src.ctr.y        0.0      ! y location for center of source (mm)
-src.ctr.z        -0.1     ! z location for center of source (mm)
-src.r            0.1      ! radius of source (mm)

-scint.nang.a1  1          ! Number of division across D mirror
-scint.nang.a2  24         ! Number of rotation angles around D mirror
-scint.sz.x       0.024    ! Size of scintilator pixel in x direction (mm)
-scint.sz.y       0.024    ! Size of scintilator pixel in y direction (mm)
-scint.sz.z       0.024    ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x      0.0      ! x location for center of scintilator (mm)
-scint.ctr.y      0.0      ! y location for center of scintilator (mm)
-scint.ctr.z      5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x  51         ! number of pixels in x direction
-scint.ngrid.y  51         ! number of pixels in y direction

-obj.sz.x         0.002    ! Size of object voxel in x direction (mm)
-obj.sz.y         0.002    ! Size of object voxel in y direction (mm)
-obj.sz.z         0.002    ! Size of object voxel in z direction (mm)
-obj.ctr.x        0.0      ! x location for center of object (mm)
-obj.ctr.y        0.0      ! y location for center of object (mm)
-obj.ctr.z        0.0      ! z location for center of object (mm)
-obj.ngrid.x      51       ! number of voxels in x direction
-obj.ngrid.y      51       ! number of voxels in y direction
-obj.ngrid.z      51       ! number of voxels in z direction

-ops.sctr         -0.05    ! location of z center for first projection
-ops.dctr         0.002    ! step size for center along z
-ops.nctr         51       ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       0.100000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.024000000000
Scintilator - y pixel size (mm) ============       0.024000000000
Scintilator - number x pixels ==============      51.000000000000
Scintilator - number y pixels ==============      51.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============       0.050000000000
Object x voxel size (mm) ===================       0.002000000000
Object y voxel size (mm) ===================       0.002000000000
Object z voxel size (mm) ===================       0.002000000000
Object number x voxels  ====================      51.000000000000
Object number y voxels  ====================      51.000000000000
Object number z voxels  ====================      51.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

**Simulation 3**

**SCT File**

```
! Sct file for wolter microscope

-ofile            test12

-ops.power_flag        0        ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x        0.0      ! x location for center of source (mm)
-src.ctr.y        0.0      ! y location for center of source (mm)
-src.ctr.z        -0.1     ! z location for center of source (mm)
-src.r            0.1      ! radius of source (mm)

-scint.nang.a1    1        ! Number of division across D mirror
-scint.nang.a2    24       ! Number of rotation angles around D mirror
-scint.sz.x       0.024    ! Size of scintilator pixel in x direction (mm)
-scint.sz.y       0.024    ! Size of scintilator pixel in y direction (mm)
-scint.sz.z       0.024    ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x      0.0      ! x location for center of scintilator (mm)
-scint.ctr.y      0.0      ! y location for center of scintilator (mm)
-scint.ctr.z      5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x    61       ! number of pixels in x direction
-scint.ngrid.y    61       ! number of pixels in y direction

-obj.sz.x         0.002    ! Size of object voxel in x direction (mm)
-obj.sz.y         0.002    ! Size of object voxel in y direction (mm)
-obj.sz.z         0.002    ! Size of object voxel in z direction (mm)
-obj.ctr.x        0.0      ! x location for center of object (mm)
-obj.ctr.y        0.0      ! y location for center of object (mm)
-obj.ctr.z        0.0      ! z location for center of object (mm)
-obj.ngrid.x      51       ! number of voxels in x direction
-obj.ngrid.y      51       ! number of voxels in y direction
-obj.ngrid.z      51       ! number of voxels in z direction

-ops.sctr         -0.05    ! location of z center for first projection
-ops.dctr         0.002    ! step size for center along z
-ops.nctr         26       ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       0.100000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.024000000000
Scintilator - y pixel size (mm) ============       0.024000000000
Scintilator - number x pixels ==============      61.000000000000
Scintilator - number y pixels ==============      61.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============       0.000000000000
Object x voxel size (mm) ===================       0.002000000000
Object y voxel size (mm) ===================       0.002000000000
Object z voxel size (mm) ===================       0.002000000000
Object number x voxels  ====================      51.000000000000
Object number y voxels  ====================      51.000000000000
Object number z voxels  ====================       1.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

## Simulation 4

### SCT File

```
! Sct file for wolter microscope

-ofile          obj13

-ops.power_flag      0       ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x      0.0     ! x location for center of source (mm)
-src.ctr.y      0.0     ! y location for center of source (mm)
-src.ctr.z      -0.1    ! z location for center of source (mm)
-src.r          0.1     ! radius of source (mm)

-scint.nang.a1  1       ! Number of division across D mirror
-scint.nang.a2  24      ! Number of rotation angles around D mirror
-scint.sz.x     0.006   ! Size of scintilator pixel in x direction (mm)
-scint.sz.y     0.006   ! Size of scintilator pixel in y direction (mm)
-scint.sz.z     0.006   ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x    0.0     ! x location for center of scintilator (mm)
-scint.ctr.y    0.0     ! y location for center of scintilator (mm)
-scint.ctr.z    5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x  101     ! number of pixels in x direction
-scint.ngrid.y  101     ! number of pixels in y direction

-obj.sz.x       0.0005 ! Size of object voxel in x direction (mm)
-obj.sz.y       0.0005 ! Size of object voxel in y direction (mm)
-obj.sz.z       0.0005 ! Size of object voxel in z direction (mm)
-obj.ctr.x      0.0     ! x location for center of object (mm)
-obj.ctr.y      0.0     ! y location for center of object (mm)
-obj.ctr.z      0.0     ! z location for center of object (mm)
-obj.ngrid.x    101     ! number of voxels in x direction
-obj.ngrid.y    101     ! number of voxels in y direction
-obj.ngrid.z    101     ! number of voxels in z direction

-ops.sctr       -0.05   ! location of z center for first projection
-ops.dctr       -0.0005         ! step size for center along z
-ops.nctr       51      ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       0.100000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.006000000000
Scintilator - y pixel size (mm) ============       0.006000000000
Scintilator - number x pixels ==============     101.000000000000
Scintilator - number y pixels ==============     101.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============      -0.025000000000
Object x voxel size (mm) ===================       0.000500000000
Object y voxel size (mm) ===================       0.000500000000
Object z voxel size (mm) ===================       0.000500000000
Object number x voxels =====================     101.000000000000
Object number y voxels  ====================     101.000000000000
Object number z voxels  ====================     101.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

### Simulation 5

### SCT File

```
! Sct file for wolter microscope

-ofile          obj14

-ops.power_flag      0        ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x      0.0     ! x location for center of source (mm)
-src.ctr.y      0.0     ! y location for center of source (mm)
-src.ctr.z      -0.1    ! z location for center of source (mm)
-src.r          0.1     ! radius of source (mm)

-scint.nang.a1  1       ! Number of division across D mirror
-scint.nang.a2  24      ! Number of rotation angles around D mirror
-scint.sz.x     0.006   ! Size of scintilator pixel in x direction (mm)
-scint.sz.y     0.006   ! Size of scintilator pixel in y direction (mm)
-scint.sz.z     0.006   ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x    0.0     ! x location for center of scintilator (mm)
-scint.ctr.y    0.0     ! y location for center of scintilator (mm)
-scint.ctr.z    5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x  101     ! number of pixels in x direction
-scint.ngrid.y  101     ! number of pixels in y direction

-obj.sz.x       0.0005 ! Size of object voxel in x direction (mm)
-obj.sz.y       0.0005 ! Size of object voxel in y direction (mm)
-obj.sz.z       0.0005 ! Size of object voxel in z direction (mm)
-obj.ctr.x      0.0     ! x location for center of object (mm)
-obj.ctr.y      0.0     ! y location for center of object (mm)
-obj.ctr.z      0.0     ! z location for center of object (mm)
-obj.ngrid.x    101     ! number of voxels in x direction
-obj.ngrid.y    101     ! number of voxels in y direction
-obj.ngrid.z    101     ! number of voxels in z direction

-ops.sctr       0.0     ! location of z center for first projection
-ops.dctr       -0.0005         ! step size for center along z
-ops.nctr       51      ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       0.100000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.006000000000
Scintilator - y pixel size (mm) ============       0.006000000000
Scintilator - number x pixels ==============     101.000000000000
Scintilator - number y pixels ==============     101.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============      -0.025000000000
Object x voxel size (mm) ===================       0.000500000000
Object y voxel size (mm) ===================       0.000500000000
Object z voxel size (mm) ===================       0.000500000000
Object number x voxels  ====================     101.000000000000
Object number y voxels  ====================     101.000000000000
Object number z voxels  ====================     101.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

### Simulation 6

### SCT File

```
! Sct file for wolter microscope

-ofile          obj15

-ops.power_flag        0        ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x      0.0     ! x location for center of source (mm)
-src.ctr.y      0.0     ! y location for center of source (mm)
-src.ctr.z      -0.1    ! z location for center of source (mm)
-src.r          3.0     ! radius of source (mm)

-scint.nang.a1  1       ! Number of division across D mirror
-scint.nang.a2  24      ! Number of rotation angles around D mirror
-scint.sz.x     0.024   ! Size of scintilator pixel in x direction (mm)
-scint.sz.y     0.024   ! Size of scintilator pixel in y direction (mm)
-scint.sz.z     0.024   ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x    0.0     ! x location for center of scintilator (mm)
-scint.ctr.y    0.0     ! y location for center of scintilator (mm)
-scint.ctr.z    5000.0  ! z location for center of scintilator (mm)
-scint.ngrid.x  501     ! number of pixels in x direction
-scint.ngrid.y  501     ! number of pixels in y direction

-obj.sz.x       0.002   ! Size of object voxel in x direction (mm)
-obj.sz.y       0.002   ! Size of object voxel in y direction (mm)
-obj.sz.z       0.002   ! Size of object voxel in z direction (mm)
-obj.ctr.x      0.0     ! x location for center of object (mm)
-obj.ctr.y      0.0     ! y location for center of object (mm)
-obj.ctr.z      0.0     ! z location for center of object (mm)
-obj.ngrid.x    501     ! number of voxels in x direction
-obj.ngrid.y    501     ! number of voxels in y direction
-obj.ngrid.z    11      ! number of voxels in z direction

-ops.sctr       0.0     ! location of z center for first projection
-ops.dctr       -0.002  ! step size for center along z
-ops.nctr       6       ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       3.000000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.024000000000
Scintilator - y pixel size (mm) ============       0.024000000000
Scintilator - number x pixels ==============     501.000000000000
Scintilator - number y pixels ==============     501.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============      -0.010000000000
Object x voxel size (mm) ===================       0.002000000000
Object y voxel size (mm) ===================       0.002000000000
Object z voxel size (mm) ===================       0.002000000000
Object number x voxels  ====================     501.000000000000
Object number y voxels  ====================     501.000000000000
Object number z voxels  ====================      11.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

## Simulation 7

### SCT File

```
! Sct file for wolter microscope

-ofile            obj16

-ops.power_flag        0         ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x        0.0      ! x location for center of source (mm)
-src.ctr.y        0.0      ! y location for center of source (mm)
-src.ctr.z        -0.1     ! z location for center of source (mm)
-src.r            3.0      ! radius of source (mm)

-scint.nang.a1 1           ! Number of division across D mirror
-scint.nang.a2 24          ! Number of rotation angles around D mirror
-scint.sz.x       0.024    ! Size of scintilator pixel in x direction (mm)
-scint.sz.y       0.024    ! Size of scintilator pixel in y direction (mm)
-scint.sz.z       0.024    ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x      0.0      ! x location for center of scintilator (mm)
-scint.ctr.y      0.0      ! y location for center of scintilator (mm)
-scint.ctr.z      5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x  501        ! number of pixels in x direction
-scint.ngrid.y  501        ! number of pixels in y direction

-obj.sz.x         0.002    ! Size of object voxel in x direction (mm)
-obj.sz.y         0.002    ! Size of object voxel in y direction (mm)
-obj.sz.z         0.002    ! Size of object voxel in z direction (mm)
-obj.ctr.x        0.0      ! x location for center of object (mm)
-obj.ctr.y        0.0      ! y location for center of object (mm)
-obj.ctr.z        0.0      ! z location for center of object (mm)
-obj.ngrid.x      501      ! number of voxels in x direction
-obj.ngrid.y      501      ! number of voxels in y direction
-obj.ngrid.z      11       ! number of voxels in z direction

-ops.sctr         0.0      ! location of z center for first projection
-ops.dctr         -0.002 ! step size for center along z
-ops.nctr         6        ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ============       0.000000000000
Source - y center location (mm) ============       0.000000000000
Source - z center location (mm) ============      -0.100000000000
Source - radius (mm)  ======================       3.000000000000

Scintilator - x center location (mm) =======       0.000000000000
Scintilator - y center location (mm) =======       0.000000000000
Scintilator - z center location (mm) =======    5000.000000000000
Scintilator - x pixel size (mm) ============       0.024000000000
Scintilator - y pixel size (mm) ============       0.024000000000
Scintilator - number x pixels ==============     501.000000000000
Scintilator - number y pixels ==============     501.000000000000
Scintilator - number ang1 steps ============       1.000000000000
Scintilator - number ang2 steps ============      24.000000000000


Object x center location (mm) ==============       0.000000000000
Object y center location (mm) ==============       0.000000000000
Object z center location (mm) ==============      -0.010000000000
Object x voxel size (mm) ===================       0.002000000000
Object y voxel size (mm) ===================       0.002000000000
Object z voxel size (mm) ===================       0.002000000000
Object number x voxels  ====================     501.000000000000
Object number y voxels  ====================     501.000000000000
Object number z voxels  ====================      11.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

**Simulation 8**

**SCT File**

```
! Sct file for wolter microscope

-ofile            obj22

-ops.power_flag         0         ! make reflection power variable (0% to 60%) (=1) or off(=0)

-src.ctr.x        0.0      ! x location for center of source (mm)
-src.ctr.y        0.0      ! y location for center of source (mm)
-src.ctr.z        -0.1     ! z location for center of source (mm)
-src.r            0.325    ! radius of source (mm)

-scint.nang.a1    1        ! Number of division across D mirror
-scint.nang.a2    24       ! Number of rotation angles around D mirror
-scint.sz.x       0.006    ! Size of scintilator pixel in x direction (mm)
-scint.sz.y       0.006    ! Size of scintilator pixel in y direction (mm)
-scint.sz.z       0.006    ! Size of scintilator pixel in z direction (mm)
-scint.ctr.x      0.0      ! x location for center of scintilator (mm)
-scint.ctr.y      0.0      ! y location for center of scintilator (mm)
-scint.ctr.z      5000.0 ! z location for center of scintilator (mm)
-scint.ngrid.x    251      ! number of pixels in x direction
-scint.ngrid.y    251      ! number of pixels in y direction

-obj.sz.x         0.0005 ! Size of object voxel in x direction (mm)
-obj.sz.y         0.0005 ! Size of object voxel in y direction (mm)
-obj.sz.z         0.0005 ! Size of object voxel in z direction (mm)
-obj.ctr.x        0.0      ! x location for center of object (mm)
-obj.ctr.y        0.0      ! y location for center of object (mm)
-obj.ctr.z        0.0      ! z location for center of object (mm)
-obj.ngrid.x      251      ! number of voxels in x direction
-obj.ngrid.y      251      ! number of voxels in y direction
-obj.ngrid.z      251      ! number of voxels in z direction

-ops.sctr         -0.0375         ! location of z center for first projection
-ops.nctr         1        ! number of projections to process

-amatrix.vfile  v0022.at ! file to save values of a-matrix
-amatrix.ifile  i0022.at ! file to save indicies of a-matrix
-amatrix.ctr    0.0    ! location of z center for start of a-matrix
-amatrix.nz     501    ! number of projections to process
```

## Output File

```
WOLTER DATA:
March 7, 2002

Source - x center location (mm) ==================       0.000000000000
Source - y center location (mm) ==================       0.000000000000
Source - z center location (mm) ==================      -0.100000000000
Source - radius (mm)  ============================       0.325000000000

Scintilator - x center location (mm) =============       0.000000000000
Scintilator - y center location (mm) =============       0.000000000000
Scintilator - z center location (mm) =============    5000.000000000000
Scintilator - x pixel size (mm) ==================       0.006000000000
Scintilator - y pixel size (mm) ==================       0.006000000000
Scintilator - number x pixels ====================     251.000000000000
Scintilator - number y pixels ====================     251.000000000000
Scintilator - number ang1 steps ==================       1.000000000000
Scintilator - number ang2 steps ==================      24.000000000000


Object x center location (mm) ====================       0.000000000000
Object y center location (mm) ====================       0.000000000000
Object z center location (mm) ====================       0.000000000000
Object x voxel size (mm) =========================       0.000500000000
Object y voxel size (mm) =========================       0.000500000000
Object z voxel size (mm) =========================       0.000500000000
Object number x voxels  ==========================     251.000000000000
Object number y voxels  ==========================     251.000000000000
Object number z voxels  ==========================     251.000000000000

mirror C - major axis - a (mm) =============     227.440417451930
mirror C - minor axis - b (mm) =============      14.580000000000
mirror C - focal length - c (mm) ===========     227.907261601530
mirror C - hyperbola center - h (mm) =======    -227.907261601530
mirror C - theta front (deg) ===============       5.600000000000
mirror C - theta back (deg) ================       5.421000000000
mirror C - theta difference (deg) ==========       0.179000000000
mirror C - z position of optic front (mm) ==     340.752820097063
mirror C - radius of optic front (mm) ======      33.411106131911
mirror C - z position of optic back (mm) ===     383.037895911492
mirror C - radius of optic back (mm) =======      36.349391658109
C-D intersection - z location (mm) =========     383.037895911492
C-D intersection - radius (mm) =============      36.349391658109

mirror D - major axis - a (mm) =============    2728.372396151007
mirror D - minor axis - b (mm) =============      50.377615866900
mirror D - focal length - c (mm) ===========    2727.907261601530
mirror D - ellipse center - h (mm) =========    2272.092738398470
mirror D - z position of optic front (mm) ==     383.037895911492
mirror D - z position of optic back (mm) ===     430.301982825933
image  x location (mm) =====================       0.000000000000
image  y location (mm) =====================       0.000000000000
image  z location (mm) =====================    5000.000000000000
system length (mm) =========================    5000.000000000000
magnification estimate (imaging) ===========      12.000000000000
```

## References

[1] Nederbragt, W W, “Wolter Instrument-Optical Design”, October 11, 2002, LLNL, Livermore, CA., UCRL-ID-150702.

[2] D. M. Goodman, E. M. Johansson, and T. W. Lawrence, “On applying the conjugate-gradient algorithm to image processing problems,” Chapter 11 in Multivariate Analysis: Future Directions, C. R. Rao, editor, Elsevier Science Publishers, 1993.